Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων
& Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας Πληροφορικης &
Υπολογιστων

Διαχείριση & Εξερεύνηση Μεγάλων Συνόλων Διασυνδεδεμένων Δεδομένων

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

της

ΜΑΡΙΑΣ Κ. ΚΡΟΜΜΥΔΑ

Διπλωματούχου Ηλεκτρολόγου Μηχανικού &
Μηχανικού Υπολογιστών, Ε.Μ.Π. (2013)

Αθήνα, Ιούνιος 2021

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ &
ΥΠΟΛΟΓΙΣΤΩΝ

Διαχείριση & Εξερεύνηση Μεγάλων Συνόλων Διασυνδεδεμένων Δεδομένων

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

της

ΜΑΡΙΑΣ Κ. ΚΡΟΜΜΥΔΑ

Διπλωματούχου Ηλεκτρολόγου Μηχανικού&
Μηχανικού Υπολογιστών, Ε.Μ.Π. (2013)

Συμβουλευτική Επιτροπή: Βασιλική Καντερέ
Άγγελος Αμδίτης
Νεκτάριος Κοζύρης

...	...	...
Β. Καντερε	Α. Αμδίτης	Ν. Κοζύρης
Επ. Καθηγήτρια Ε.Μ.Π.	Ερευνητής Α' Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.
...	...	...
Θ. Βαρβαρίγου	Σ. Παπαβασιλείου	Δ. Χατζηαντωνίου
Καθηγήτρια Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Αν. Καθηγητής Ο.Π.Α.
...		
Β. Καρυώτης		
Αν. Καθηγητής Ιόνιο Παν.		

Αθήνα,24 Ιούνιος 2021

. . .

ΜΑΡΙΑ Κ. ΚΡΟΜΜΥΔΑ
Υποψήφια Διδάκτωρ Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

National Technical University of Athens

School of Electrical
& Computer Engineering

Management & Exploration of Big Linked Datasets

Doctoral Thesis

of

MARIA K. KROMMYDA

Diploma from School of Electrical &
Computer Engineering, NTUA, 2013

Athens, June 2021

National Technical University of Athens
School of Electrical & Computer Engineering

Management & Exploration of Big Linked Datasets

Doctoral Thesis
of

MARIA K. KROMMYDA

Diploma from School of Electrical &
Computer Engineering, NTUA, 2013

Advisors' Board: Verena Kantere
Angelos Amditis
Nektarios Koziris

. . .
V. Kantere
As. Professor NTUA

. . .
A. Amditis
Researcher A'

. . .
N. Koziris
Professor NTUA

. . .
Th. Varvarigou
Professor NTUA

. . .
S. Papavassiliou
Professor NTUA

. . .
D. Xatziantoniou
As. Professor AUEB

. . .
V. Karyotis
As. Professor Ionian Un.

Athens,24 June 2021

. . .

MARIA K. KROMMYDA
Candidate Doctor of School of Electrical & Computer Engineering, NTUA

ΠΡΟΛΟΓΟΣ

Λέξεις κλειδιά:Διαχείριση δεδομένων, οπτικοποίηση, μεγάλα σύνολα δεδομένων*Η παρακάτω διατριβή παρουσιάζει την ερευνητική μελέτη πάνω σε ζητήματα διαχείρισης και εξερεύνησης μεγάλων συνόλων διασυνδεδεμένων δεδομένων. Στα πλαίσια της διδακτορικής διατριβής έχει ερευνηθεί σε βάθος τόσο η υπάρχουσα βιβλιογραφία όσο και η σχετική ερευνητική εργασία σε παγκόσμιο επίπεδο. Έχουν διερευνηθεί ενδελεχώς τα ζητήματα αυτά τόσο από την πλευρά των μηχανικών που καλούνται να σχεδιάσουν συστήματα που να διαχειρίζονται τα χαρακτηριστικά αυτών των δεδομένων, όσο και από την πλευρά των χρηστών που επιθυμούν ομαλή και ανεμπόδιστη πρόσβαση στα δεδομένα με εύκολους, ως προς την χρήση και την κατανόηση, τρόπους. Επιπλέον, έχουν προταθεί πλήρεις λύσεις για την αντιμετώπιση αυτών των ζητημάτων με βάση τα σενάρια χρήσης.*

Συγκεκριμένα, έχει προταθεί ένα ολοκληρωμένο σύστημα οπτικοποίησης της πληροφορίας βασισμένη σε χαρακτηριστικά του SPARQL ερωτήματος. Η προτεινόμενη λύση περιλαμβάνει ένα σύστημα υποστήριξης λήψεων αποφάσεων που συμβάλει στην επιλογή της κατάλληλης οπτικοποίησης για κάθε ερώτημα SPARQL που μπορεί να δημιουργήσει ο χρήστης, βασισμένο σε μια βάση γνώσεων που περιλαμβάνει τα αποτελέσματα μια εκτεταμένης πειραματικής μελέτης κατά την οποία αναλύθηκαν συγκεκριμένα χαρακτηριστικά πολλών SPARQL συνόλων δεδομένων.

Προτείνεται ακόμα μια λύση η οποία στοχεύει στο να βοηθήσει χρήστες που δεν είναι εξοικειωμένοι με τα μεγάλα σύνολα δεδομένων και τον Σημασιολογικό Ιστό στο να εξερευνήσουν σύνολα δεδομένων τα οποία δεν ενημερώνονται συχνά αλλά περιέχουν σημαντικές πληροφορίες που πρέπει να εξερευνηθούν σε βάθος.

Για την αξιοποίηση των συνόλων δεδομένων ζευγών ερώτηση-απάντηση που είναι διαθέσιμα με τέτοιο τρόπο που να εξαλείφονται τα μη-αξιοποιήσιμα και υποκειμενικά δεδομένα σε συστήματα αυτόματων διαλόγων, αναπτύχθηκαν τεχνικές σημασιολογικής ανάλυσης των δεδομένων. Προτάθηκε μια τεχνική που ορίζει μια αυστηρή ροή δεδομένων και εξασφαλίζει ότι τα σύνολα δεδομένων που δίνονται ως είσοδο επεξεργάζονται με τον καλύτερο δυνατό τρόπο τόσο με βάση τον σημασιολογικό προσανατολισμό του συστήματος όσο και με βάση την περίπτωση χρήσης.

Όπως είναι αναμενόμενο σε κάθε μεγάλο σύνολο δεδομένων έτσι και για δεδομένα που συλλέγονται από τους πολίτες η ποιότητα και η αξιοπιστία των μετρήσεων που συλλέγονταί είναι αμφισβητούμενη. Για τον λόγο αυτόν αναπτύχθηκε ένας μηχανισμός ελέγχου της ποιότητας των δεδομένων που βασίστηκε σε μια σειρά από κανόνες και πρακτικούς περιορισμούς.

Λέξεις κλειδιά:Διαχείριση δεδομένων, οπτικοποίηση, μεγάλα σύνολα δεδομένων

Μαρία Κρομμύδα
Αθήνα, Ιούνιος 2021

ABSTRACT

This document presents the research contribution regarding the exploration and visualization of very large linked datasets. First, the technologies and innovations that led to the increase of the available big data are discussed. Then the challenges that people interested in the exploration and analysis of the available information are discussed. Emphasis is given in differentiating the challenges related to the nature and characteristics of the available dataset from the ones coming from specific use cases and target audience. Specific, real-world examples are presented to show the needs of the users and the specification for the solutions.

Next, a solution that supports users with querying SPARQL endpoints, visualizing the results, in the optimal way based on a knowledge base and a decision support system, and facilitating the exploration of the information through an innovative functionality toolkit is presented. The solution is proposing a client-server architectural model, that allows the users to perform SPARQL queries over any available endpoint, receive the results visualized based on the specific characteristics of the query and explore the visualized information through multiple abstraction and filtering criteria.

In addition, a fully-fledged innovative system that supports the representation of any RDF dataset as one continuous graph at the two-dimensional space. The system has been carefully designed to manage any dataset independently of its specific characteristics. The system stores the information in a distributed key-value storage system and indexes the information with a XZ-index ensuring the smooth and timely provision of the information to multiple users regardless the spatial criteria used or the area requested. A dedicated user interface, allows the user to access the information, explore the complete graph, visualize the dataset thought multiple abstraction and filtering criteria, navigate paths of interest or isolate parts of the dataset that wants to further explore.

Understanding that the value of the available dataset is closely related to their quality, a technique to improve the quality of the available conversational datasets is proposed. The technique builds on top of semantic relationships, such as synonyms and hyponomy, to calculate the semantic similarity and the semantic relatedness between the topic that the dataset is to be used for and the available information. Taking into consideration the use case that the output dataset is going to be used for, its thematic relation with the source of the input dataset and the language formality needed for the task, the two scores are merged using a weight-based score function into a matching percentage. The dataset is then ranked based on this percentage and only the information above the required threshold is present in the output file. Extended experimental analysis showed that machine learning solutions perform better when trained with smaller but properly created dataset than when trained over complete initial dataset.

Finally, the data quality control needed when collecting big datasets is discussed. The specific example of the data collected within the context of the SCENT EU founded project is presented. There volunteers were tasked to use mobile applications and smart sensor to collect images, video and sensor measurement at area of hydrological interest. The collected data were processed in order to collect information about the land cover of the area, the water level and the water velocity of the water body as well as air temperature and soil moisture values. The data were collected from volunteers with no training regarding the proper way to collect scientific measurements, in conditions that were challenging regarding the weather phenomena and the accessibility and in areas that had many technological challenges such as the lack of accurate GPS signal. The collected data are to be used in order to update hydrological models, meaning that there is a need for high accuracy in the measurements used. Innovative techniques that filter out invalid measurements were developed in order to provide the proper data for the models. The techniques were proven to work properly and they were able to support the creation of improved, more accurate flood models.

Keywords: Data management, big data, data visualization

Contents

List of Figures

ΠΕΡΙΛΗΨΗ

Ήταν πριν από περίπου είκοσι χρόνια που ο Tim Berners-Lee παρουσίασε την ιδέα του Σημασιολογικού Ιστού, όταν του ζητήθηκε να ενημερώσει τον τηλεφωνικό κατάλογο που χρησιμοποιούνταν στο ερευνητικό κέντρο CERN. Χιλιάδες ερευνητές και εκατοντάδες έργα, άτομα που εργάζονταν μόνο για μερικούς μήνες, συχνές αλλαγές γραφείων και εργασιών αλλά και άτομα που μοίραζαν τον χρόνο τους ανάμεσα σε δύο ή περισσότερες εργασίες έκαναν μια εργασία που έμοιαζε αρχικά απλή να εξελιχθεί σε χαώδης. Με βάση τις παρατηρήσεις του κατά την δημιουργία του τηλεφωνικού καταλόγου ο Berners-Lee επικεντρώθηκε στην ανάγκη δημιουργίας ενός συστήματος που να προσφέρει κεντρική αποθήκευση της πληροφορίας η οποία θα είναι προσβάσιμη σε πολλούς χρήστες και θα συνοδεύεται από τα απαραίτητα εργαλεία σημασιολογικής κατανόησης και συσχέτισης.

Η ιδέα, αρχικά, δεν ήταν εύκολο να υλοποιηθεί σε ήδη υπάρχοντα σύνολα δεδομένων αφου απαιτούσε από τους ερευνητές να αφιερώσουν πολύ χρόνο και προσπάθεια ώστε να μετατρέψουν τα δεδομένα στην κατάλληλη μορφή και να δημοσιευθούν κεντρικά με την κατάλληλη σημασιολογική επισήμανση. Καθώς το ενδιαφέρον για τον Σημασιολογικό Ιστό αυξανόταν, δημιουργήθηκαν τα πρώτα εργαλεία που αυτοματοποιούσαν αυτήν την διαδικασία, με αποτέλεσμα να γίνει πιο προσβάσιμη και λιγότερο απαιτητική σε πόρους και να κερδίσει σε δημοτικότητα.

Όπως ήταν αναμενόμενο, τα επόμενα χρόνια μετά την αυτοματοποίηση της διαδικασίας, ο Σημασιολογικός Ιστός αναπτύχθηκε και επεκτάθηκε πολύ γρήγορα καθώς οι περισσότεροι παραγωγοί δεδομένων αναγνώρισαν το πόσο σημαντικό είναι να υπάρχει ένας κοινός τρόπος διαμοιρασμού πληροφορίας, που θα συμβάλει στην ανάπτυξη και την βιωσιμότητα των προϊόντων τους. Μέσα σε λίγα χρόνια, παγκόσμιοι οργανισμοί, εγκυκλοπαίδειες και επιστημονικές βάσεις δεδομένων έγιναν μέρος του σημασιολογικού ιστού προσφέροντας τα δεδομένα τους με την χρήση του Πλαισίου Περιγραφής Πόρων (Resource Description Framework).

Πλαίσιο Περιγραφής Πόρων. *Το Πλαίσιο Περιγραφής Πόρων είναι ένα μοντέλο ανταλλαγής πληροφορίας που έχει ως στόχο να συνενώσει ετερογενείς πηγές δεδομένων ακόμα και όταν περιγράφονται με διαφορετικά μοντέλα δεδομένων και να επιτρέψει την αλλαγή του μοντέλου δεδομένων χωρίς να επηρεάζονται οι χρήστες τις πληροφορίας. Το Πλαίσιο Περιγραφής Πόρων χρησιμοποιεί την δομή του Διαδικτύου, τις σημασιολογικές σχέσεις ανάμεσα σε έννοιες, ώστε να δημιουργήσει μια αυστηρή δομή ενώνοντας έννοιες, οι οποίες ονομάζονται οντότητες στα πλαίσια του μοντέλου, με σημασιολογικές σχέσεις, οι οποίες ονομάζονται συσχετίσεις στα πλαίσια του μοντέλου.*

Το κυρίαρχο χαρακτηριστικό σε ότι αφορά το Πλαίσιο Περιγραφής Πόρων είναι η ευελιξία του, αφού έχει σχεδιαστεί για να επιτρέπει τη συνένωση δομημένων και ημι-

δομημένων δομών δεδομένων, από πληθώρα πηγών με τρόπο που να διευκολύνει την συνένωση της πληροφορίας και την συνολική της επεξεργασία. Αυτό το χαρακτηριστικό είναι που το καθιστά πολύ εύκολο να χρησιμοποιηθεί από υπολογιστικά συστήματα και ταυτόχρονα τόσο δύσκολο να παρουσιαστεί με έναν απλό και κατανοητό τρόπο σε ανθρώπους, και ειδικά σε χρήστες που δεν είναι εξοικειωμένοι με το μοντέλο και την δομή του. Αξίζει να επισημανθεί ότι το Πλαίσιο Περιγραφής Πόρων δεν έχει σχεδιαστεί για να απευθύνεται σε ανθρώπους και η δομή του είναι τέτοια που καθιστά την κατανόηση συνόλων δεδομένων που το ακολουθούν πολύ δύσκολη χωρίς επιπρόσθετη πληροφορία, ανάλυση και επεξεργασία.

Ο στόχος του Πλαισίου Περιγραφής Πόρων ήταν να επιτρέψει σε μηχανές να ανταλλάξουν δεδομένα και πληροφορίες με τέτοιον τρόπο ώστε να μην χρειάζεται η παρέμβαση του ανθρώπου για να γίνει κατανοητό το περιεχόμενο της πληροφορίας που ανταλλάχθηκε και να μπορεί να αξιοποιηθεί άμεσα και αποτελεσματικά. Για παράδειγμα, δύο συστήματα που καταγράφουν μετρήσεις από αισθητήρες ποιότητας αέρα, το ένα δίνοντας ειδοποιήσεις για την υπερβολική συγκέντρωση ρύπων και το άλλο για την παρουσία σωματιδίων, μπορούν να αποκτήσουν το ένα πρόσβαση στα δεδομένα του άλλο, αν αυτά είναι διαθέσιμα με βάση το Πλαίσιο Περιγραφής Πόρων, και να δώσουν τις ίδιες ειδοποιήσεις και για τα νέα δεδομένα, χωρίς καμία ανθρώπινη παρέμβαση. Το πλήθος των συνόλων δεδομένων και το θεματικό τους εύρος είναι αυτό που έχει κάνει πολλούς ερευνητές να ενδιαφέρονται για την πληροφορία που είναι διαθέσιμη και έχει οδηγήσει στην δημιουργία πολλών εργαλείων οπτικοποίησης και ανάλυσης δεδομένων που είναι διαθέσιμα με βάση το Πλαίσιο Περιγραφής Πόρων.

SPARQL Protocol and RDF Query Language. *Το γεγονός ότι το Πλαίσιο Περιγραφής Πόρων είναι αρκετά ευέλικτο σε συνδυασμό με το ότι απευθύνεται κυρίως σε μηχανές δημιουργεί ένα σημαντικό κενό σχετικά με το πως οι άνθρωποι που ενδιαφέρονται για την διαθέσιμη πληροφορία θα μπορούν να έχουν πρόσβαση σε αυτήν. Το κενό αυτό έρχεται να καλύψει η SPARQL Protocol and RDF Query Language, μια γλώσσα ερωτημάτων σχεδιασμένη ειδικά για την αναζήτηση πάνω σε δεδομένα που παρουσιάζονται από το Πλαίσιο Περιγραφής Πόρων. Η SPARQL γλώσσα ερωτημάτων έχει γίνει η βασική γλώσσα που χρησιμοποιείται από τα περισσότερα συστήματα που αποθηκεύουν τα δεδομένα τους χρησιμοποιώντας το Πλαίσιο Περιγραφής Πόρων και το 2008 καταχωρήθηκε σαν μια επίσημη πρόταση του World Wide Web Consortium.*

Η χρήση της SPARQL γλώσσας ερωτημάτων έχει δύο βασικές δυσκολίες. Η πρώτη είναι ότι για να είναι σε αρμονία με την ευελιξία του Πλαισίου Περιγραφής Πόρων η γλώσσα είναι αρκετά περίπλοκη, με ένα ευέλικτο συντακτικό, και απαιτεί αρκετή μελέτη για την δημιουργία ακόμα και απλών ερωτημάτων. Αυτό δημιουργεί συχνά προβλήματα σε χρήστες που δεν είναι εξοικειωμένοι με το Πλαίσιο Περιγραφής Πόρων και την SPARQL γλώσσα ερωτημάτων, αποθαρρύνοντας τους από το να συνεχίσουν την εξερεύνηση των συνόλων δεδομένων που τους ενδιαφέρουν ή και αποκρύπτοντας τους πληροφορία που μπορεί να είναι σημαντική για αυτούς αλλά όχι ξεκάθαρο στο πως μπορεί να εντοπιστεί. Η δεύτερη δυσκολία έγκειται στο γεγονός ότι τα δεδομένα που είναι διαθέσιμα για εξερεύνηση δεν ακολουθούν κανένα μοντέλο δεδομένων, οι περισσότεροι χρήστες όμως έχουν συνηθίσει να χρησιμοποιούν δεδομένα που βασίζονται σε συγκεκριμένα και πολύ αυστηρά μοντέλα και να δημιουργούν ερωτήματα με βάση αυτά. Αυτό το χαρακτηριστικό έχει σαν αποτέλεσμα η πολυπλοκότητα των ερωτημάτων να αυξάνεται εκθετικά και πολύ συχνά να απαιτείται μεγάλη εξοικείωση με το συντακτικό της γλώσσας ώστε να δημιουργηθούν τα κατάλληλα *ερωτήματα που θα μπορέσουν να εντοπίσουν τη ζητούμενη πληροφορία.*

Ένα τελικό σημείο σύνδεσης (endpoint) SPARQL μπορεί να οριστεί σαν ένα σημείο εξυπηρέτησης ερωτημάτων που ακολουθούν την SPARQL γλώσσα, το οποίο είναι διαθέσιμο πάνω σε ένα απομακρυσμένο σύστημα αποθήκευσης και διαχείρισης δεδομένων που ακολουθούν το μοντέλο του Πλαισίου Περιγραφής Πόρων. Το τελικό σημείο σύνδεσης SPARQL επιτρέπει στους χρήστες μέσω απλών αιτημάτων που ακολουθούν το HTTP πρωτόκολλο επικοινωνίας, είτε είναι άνθρωποι είτε είναι μηχανές, να θέσουν ερωτήματα στα διαθέσιμα σύνολα δεδομένων και να λάβουν τις απαντήσεις σε κάποια φιλική προς τις μηχανές μορφή, όπως είναι τα SPARQL Query Results XML και JSON μοντέλα δεδομένων που έχουν δημιουργηθεί για την μοντελοποίηση των αποτελεσμάτων SPARQL ερωτημάτων. Πρακτικά, ένα τελικό σημείο σύνδεσης SPARQL μπορεί να χαρακτηριστεί μια φιλική προς τις μηχανές διεπαφή πάνω σε μια βάση γνώσεων του Πλαισίου Περιγραφής Πόρων. Η πρόσβαση στην πληροφορία, η δημιουργία των κατάλληλων ερωτημάτων η κατανόηση και η αναπαράσταση της πληροφορίας που επιστρέφεται είναι ευθύνη του χρήστη του τελικού σημείου σύνδεσης SPARQL και όχι αυτού που προσφέρει το τελικό σημείο σύνδεσης SPARQL.

Καθώς ο αριθμός και η θεματολογία των συνόλων δεδομένων που είναι διαθέσιμα αυξάνεται, τόσο αυξάνεται και το ενδιαφέρον επιστημών και ερευνητών από διάφορες επιστημονικές κοινότητες να χρησιμοποιήσουν την διαθέσιμη πληροφορία. Οι νέοι χρήστες αυτών των δεδομένων είναι συνήθως άνθρωποι με πολύ περιορισμένες γνώσεις για τον Σημασιολογικό Ιστό, αδυναμία συγγραφής των σωστών ερωτημάτων και περιορισμένο ενδιαφέρον για συγκεκριμένα σύνολα δεδομένων.

Για να ανταποκριθούν στις ανάγκες αυτών των χρηστών, κάποια τελικά σημεία σύνδεσης SPARQL αναβαθμίστηκαν ώστε να προσφέρουν στους χρήστες πρόσβαση στην πληροφορία μέσα από μηχανές αναζήτησης και να την παρουσιάζουν με δομημένους τρόπους. Αυτές οι προσπάθειες είναι περιορισμένες και διαθέσιμες μόνο για κάποια τελικά σημεία σύνδεσης SPARQL και προσφέρουν πολύ περιορισμένες δυνατότητες εξερεύνησης και οπτικοποίησης της πληροφορίας. Οι δομημένοι τρόποι παρουσίασης της πληροφορίας, συνήθως λίστες και πίνακες, παρουσιάζουν την πληροφορία σε μια μορφή που μπορεί να διαβαστεί από τους ανθρώπους αλλά χωρίς να δείχνουν σχέσεις και συσχετίσεις ανάμεσα σε οντότητες που είναι σημαντικές για την κατανόηση της πληροφορίας.

Linked Open Data. *Σε συνέχεια της προσπάθειας του Σημασιολογικού Ιστού, ήρθε η πρωτοβουλία για το Linked Open Data, μια προσπάθεια που έκανε διαθέσιμα πολλά διασυνδεδεμένα σύνολα δεδομένων που έχουν την δυνατότητα να προσφέρουν πολύτιμες γνώσεις. Και σε αυτήν την περίπτωση όμως η πρόσβαση στα δεδομένα, η κατανόηση τους και η αναπαράσταση του με φιλικό για τον χρήστη τρόπο έχει πολλές προκλήσεις. Πρόσφατα έχουν γίνει πολλές προσπάθειες να βρεθούν τεχνικές για την επεξεργασία αυτών των δεδομένων, την εξαγωγή της πληροφορίας που περιέχουν και την παρουσίαση τους στους χρήστες με τον καλύτερο δυνατό τρόπο.*

Συνήθως τέτοια σύστημα χρειάζονται ολική ή μερική επισήμανση της πληροφορίας με την κατάλληλη σημασιολογία. Τα συστήματα εκμεταλλεύονται αυτές τις επισημάνσεις ώστε να περιορίζουν τον όγκο των δεδομένων που διαχειρίζονται. Βασισμένα στην ιεραρχική δομή της σημασιολογικής επισήμανσης, προσφέρουν στο χρήστη την δυνατότητα να εξερευνήσουν την πληροφορία με την βοήθεια των σχέσεων των οντοτήτων και να περιηγηθούν στις ιδιότητες τους. Για να είναι αποδοτικά, τέτοια συστήματα χρειάζονται σύνολα δεδομένων που να έχουν περιορισμένο αριθμό σημα-

σιολογικών κλάσεων ενώ τις περισσότερες φορές αγνοούν πλήρως πληροφορίες που δεν έχουν επισημανθεί σημασιολογικά.

Άλλα συστήματα επιλέγουν να δημιουργήσουν ομάδες αντικειμένων που ανήκουν στην ίδια σημασιολογική κατηγορία και να παρουσιάσουν στους χρήστες αυτές τις ομάδες μέσα από βασικά διαγράμματα οπτικοποίησης. Κάποια από αυτά τα συστήματα επιλέγουν να οπτικοποιήσουν μόνο συγκεκριμένους τύπους δεδομένων ενώ άλλα δίνουν την ευελιξία στους χρήστες να επιλέξουν την πληροφορία που τους ενδιαφέρει να οπτικοποιήσουν.

Πολλά συστήματα επιλέγουν να παρουσιάσουν την πληροφορία με την μορφή γράφων καθοδηγώντας τους χρήστες να ακολουθήσουν τις σχέσεις ανάμεσα στις οντότητες ώστε να αποκτήσουν πρόσβαση στην πληροφορία που τους ενδιαφέρει. Τα συστήματα αυτά επιτρέπουν στον χρήστη να ξεκινήσει την εξερεύνηση της πληροφορίας από κάποια συγκεκριμένη οντότητα, μέσα από την αναζήτηση με λέξεις-κλειδιά ή και με βάση των τύπο των οντοτήτων που τον ενδιαφέρουν.

Τέλος, κάποια συστήματα στην προσπάθεια τους να αποφύγουν την ανάγκη παρουσίας σημασιολογικών επισημάνσεων, επικεντρώνονται στο να παρουσιάσουν την πληροφορία με βάση την ιεραρχία των σχέσεων ανάμεσα στις οντότητες. Τέτοια συστήματα είναι πολύ δημοφιλή αφου μπορούν να αντιμετωπίσουν πολύ εύκολα σύνολα δεδομένων ανεξαρτήτως του όγκου και των χαρακτηριστικών τους, περιορίζουν όμως αρκετά την πληροφορία που παρουσιάζουν στον χρήστη με αποτέλεσμα να μην καλύπτουν πάντα τις ανάγκες εξερεύνησης.

Big Data. *Σε συνέχεια της πρωτοβουλίας για το Linked Open Data, άρχισε να χρησιμοποιείται ο όρος Big Data, για να χαρακτηρίσει μεγάλα σύνολα διασυνδεδεμένων δεδομένων, ένας όρος που κέρδισε αμέσως σε δημοτικότητα και άρχισε να χρησιμοποιείται σε πολλούς ερευνητικούς τομείς και επιστημονικές εργασίες. Μια πιο προσεχτική ματιά στην χρήση του, όμως, αποκαλύπτει ότι χρησιμοποιείται πάρα πολύ συχνά με διαφορετικό νόημα και σκοπό. Υπάρχουν πολλοί ασυνεπείς ορισμοί για τον όρο, που διαφοροποιούνται έντονα τόσο ανάλογα με την χρονική στιγμή που γράφτηκαν όσο και ανάλογα το επιστημονικό πεδίο το οποίο αφορούσαν. Αυτό που είναι κοινό για όλους τους ορισμούς είναι κάποια συγκεκριμένα χαρακτηριστικά τους που αφορούν όχι μόνο τα ίδια τα σύνολα δεδομένων αλλά και τους τρόπους χρήσης τους.*

Για να χαρακτηριστεί, λοιπόν, ένα σύνολο δεδομένων ως μεγάλο, Big Data, θα πρέπει να υπακούει σε κάποιους κανόνες και να συμμορφώνεται με συγκεκριμένες προϋποθέσεις. Αναλυτικά, ένα σύνολο δεδομένων είναι μεγάλο όταν έχει τα ακόλουθα χαρακτηριστικά:

- *Όγκος Δεδομένων. Το πρώτο βασικό χαρακτηριστικό είναι ο όγκος των δεδομένων που ανήκουν στο σύνολο δεδομένων. Αν και αρχικά κάτι τέτοιο φαίνεται αναμενόμενο και προφανές, υπάρχει μεγάλη διαμάχη σχετικά με το πως ορίζεται ο μεγάλος όγκος δεδομένων. Χαρακτηριστικό παράδειγμα είναι η μετάβαση από εξωτερικές συσκευές αποθήκευσης δεδομένων που δεν ξεπερνούσαν τις μερικές δεκάδες MBs σε συσκευές με αρκετά TBs. Για να αποφευχθεί αυτή η σχετικότητα σε απόλυτους αριθμούς, ως μεγάλος όγκος ορίζεται αυτός που δεν είναι επεξεργάσιμος στην διαθέσιμη μνήμη, σε σχέση κάθε φορά με τον χρήστη της πληροφορίας και την ικανότητα του να δεχθεί και να αποθηκεύσει την πληροφορία.*

- *Ποικιλία. Το επόμενο χαρακτηριστικό των μεγάλων συνόλων δεδομένων είναι ότι προέρχονται από περισσότερες από μία πηγές και έχουν περισσότερα από ένα μοντέλα δεδομένων. Τα σύνολα δεδομένων μπορούν να περιέχουν διαφορετικούς τύπους δεδομένων όπως εικόνες και βίντεο ή και μετα-δεδομένα.*

- *Ταχύτητα. Το επόμενο χαρακτηριστικό των μεγάλων συνόλων δεδομένων είναι η ταχύτητα με την οποία δημιουργούνται. Χαρακτηριστικό παράδειγμα είναι τα κοινωνικά δίκτυα όπου κάθε λεπτό δημοσιεύονται χιλιάδες μηνύματα, εκατοντάδες φωτογραφίες και ώρες βίντεο. Η ταχύτητα με την οποία παράγονται τα δεδομένα επιβάλει και τον ρυθμό που αυτά θα πρέπει να επεξεργάζονται.*

- *Εγκυρότητα. Τα μεγάλα σύνολα δεδομένων συλλέγονται από πολλές πηγές, η κάθε μία με διαφορετική αξιοπιστία και αποτελούνται από διαφορετικά σύνολα δεδομένων τα οποία μπορεί να μην είναι πάντα αξιόπιστα. Χαρακτηριστικό παράδειγμα είναι σύνολα δεδομένων από συστήματα αισθητήρων που μπορεί να περιλαμβάνουν δεδομένα από αισθητήρες με διαφορετικό βαθμό εμπιστοσύνης ανάλογα με την αξιοπιστία των αισθητήρων και την συχνότητα συντήρησης τους.*

- *Αξία. Ένα από τα σημαντικότερα χαρακτηριστικά των μεγάλων συνόλων δεδομένων είναι η αξία που μπορεί να κερδίσει κανείς από την επεξεργασία τους. Η ανάγκη να προστεθεί αυτό το χαρακτηριστικό στην λίστα δεν υπήρχε αρχικά, όταν τα διαθέσιμα σύνολα δεδομένων προέρχονταν από αξιόπιστες πηγές που είχαν συγκεκριμένες χρήσεις, όπως εγκυκλοπαίδειες και επιστημονικές βάσεις δεδομένων γιατί θεωρείτο προφανείς. Καθώς όμως ξεκίνησε η παραγωγή μεγάλων συνόλων δεδομένων που δεν είχαν να προσφέρουν κάτι με την ετεροχρονισμένη επεξεργασία τους, όπως κάμερες ασφαλείας και αισθητήρες έξυπνων σπιτιών, προέκυψε και η ανάγκη να τονιστεί αυτό το χαρακτηριστικό.*

Η εξερεύνηση τέτοιων δεδομένων περιπλέκεται ακόμα περισσότερο αν εισάγουμε την έννοια της σημασιολογικής ομοιότητας στα κριτήρια εξερεύνησης. Η σημασιολογική ανάλυση της πληροφορίας βασίζεται σε μια σειρά από σχέσεις, οι οποίες έχουν οριστεί με βάση λεξιλογικούς κανόνες. Σε αυτούς τους κανόνες περιλαμβάνονται έννοιες όπως τα συνώνυμα, που αναφέρονται σε λέξεις που είναι ναι μεν λεξικολογικά διαφορετικές μεταξύ τους πλην όμως παρουσιάζουν την ίδια περίπου σημασία. Περιλαμβάνονται ακόμα τα αντώνυμα, λέξεις που βρίσκονται σε μια εγγενώς ασύμβατη δυαδική σχέση, όπως τα αντίθετα ζεύγη αλλά και έννοιες που τονίζουν την ιεραρχία ανάμεσα σε λέξεις όπως γενίκευση, εξειδίκευση, μέρος ενός συνόλου και ταυτόσημα.

Όλα τα σύνολα δεδομένων που έχουμε συζητήσει έως τώρα, έχουν προκύψει με την άμεση ή έμμεση παρέμβαση του ανθρώπου. Σε μια προσπάθεια να αυτοματοποιηθεί τόσο η παραγωγή των δεδομένων όσο και η εξαγωγή γνώσης από αυτά, το 1999 ο Kevin Ashton, συνιδρυτής και διευθύνων σύμβουλος της εταιρίας Auto-ID Labs, πρότεινε το διαδίκτυο των πραγμάτων (IoT). Η βασική ιδέα πίσω από το διαδίκτυο των πραγμάτων ήταν η χρήση τεχνολογιών όπως η ταυτοποίηση μέσω ραδιοσυχνότητας (Radio-frequency identification) και τα ηλεκτρομαγνητικά πεδία, που θα επιτρέπουν την αναγνώριση και παρακολούθηση εμπορευμάτων, που θα φέρουν συγκεκριμένη συσκευή ανίχνευσης, όταν βρίσκονται σε συγκεκριμένα σημεία ελέγχου. Μια τέτοια λύση είναι πολύ χαμηλού κόστους και επιτρέπει την αυτόματη παραγωγή πληροφορίας, με την παρακολούθηση των αγαθών σε όλη την διαδικασία παραγωγής και διανομής.

Ένα τέτοιο σύνολο δεδομένων προσφέρει πολλές ευκαιρίες ανάλυσης και επεξεργασίας για την εξαγωγή γνώσεων τόσο σε ότι αφορά πιθανά προβλήματα και καθυστερήσεις στην διαχείριση της παραγωγής όσο και σχετικά με ευκαιρίες βελτιστοποίησης της παραγωγής και μη ιδανικής λύσης των διαθέσιμων πόρων.

Ορισμός προβλήματος. *Το πρόβλημα της εξερεύνησης και της οπτικοποίησης συνόλων δεδομένων σαν και αυτά που περιεγράφηκαν παραπάνω έχει δύο σημαντικές κατηγορίες προκλήσεων. Η πρώτη κατηγορία αφορά προκλήσεις που σχετίζονται με τα ίδια τα δεδομένα και τα χαρακτηριστικά τους, ενώ η δεύτερη κατηγορία αφορά προκλήσεις που προέρχονται από τις λειτουργικότητες και τα σενάρια χρήσης που είναι απαραίτητα για την δημιουργία ενός συστήματος φιλικό προς τον χρήστη.*

Αρχικά, παρουσιάζονται οι προκλήσεις που αφορούν τα δεδομένα και τα χαρακτηριστικά τους, δίνοντας έμφαση στις τεχνικές δυσκολίες που προκύπτουν από αυτές, καθώς και το πως σχετίζονται με σχεδιαστικές επιλογές για την κατάλληλη αρχιτεκτονική. Αυτές είναι:

- *Όγκος δεδομένων. Ορισμένα από τα διαθέσιμα σύνολα δεδομένων είναι πάρα πολύ μεγάλα ή παράγονται με πολύ μεγάλους ρυθμούς. Ένα χαρακτηριστικό τέτοιο παράδειγμα είναι η Wikipedia, η οποία παρέχει όλη την πληροφορία που είναι διαθέσιμη στην ψηφιακή της έκδοση και ως ένα μεγάλο σύνολο δεδομένων με την χρήση του Πλαισίου Περιγραφής Πόρων. Συνήθως είναι απαραίτητο τα συστήματα που χρησιμοποιούν αυτήν την πληροφορία να την επεξεργαστούν και να την παρουσιάσουν σαν ένα ενιαίο σύνολο, κάτι που απαιτεί προσεχτικό σχεδιασμό, κατανεμημένη επεξεργασία και συνολικό κατανεμημένο αρχιτεκτονικό σχεδιασμό.*

- *Αξιολόγηση δεδομένων. Τα μεγάλα σύνολα δεδομένων αναμένονται να έρχονται από πολλές πηγές και να περιέχουν δεδομένα που διαφέρουν σε ότι αφορά την ποιότητα και την αξιοπιστία τους. Για αυτόν τον λόγο υπάρχει μεγάλη ανάγκη για πολλαπλούς ποιοτικούς ελέγχους σε όλα τα στάδια της επεξεργασίας τους και την χρήση τους μόνο μετά από προσεκτική επιβεβαίωση τους. Στην συνέχεια, θα συζητηθούν εκτενώς συστήματα που αγνόησαν την ανάγκη ελέγχου και αξιολόγησης των δεδομένων εισόδων και οδηγήθηκαν στην αποτυχία καθώς και πως τεχνικές βελτίωσης των δεδομένων εισόδου μπορούν να συμβάλουν αποτελεσματικά στην βελτίωση της συνολικής απόδοσης του συστήματος που τα χρησιμοποιεί.*

- *Ομογενοποίηση δεδομένων. Η πολλαπλότητα των πηγών εκτός από την αύξηση του κινδύνου ως προς την ποιότητα των δεδομένων, εισάγει και την πιθανότητα τα δεδομένα να μην υπακούν στο ίδιο μοντέλο, να μην περιέχουν την ίδια πληροφορία ή ακόμα και να μην χρησιμοποιούν τις ίδιες μονάδες μετρήσεις. Είναι πολύ σημαντικό, επομένως, πριν την χρήση μεγάλων συνόλων δεδομένων να λαμβάνονται όλα τα απαραίτητα μέτρα για την ομογενοποίηση τους. Δεδομένα που δεν συνοδεύονται από την κατάλληλη μετα-πληροφορία θα πρέπει να απορρίπτονται και να μην χρησιμοποιούνται για την ανάλυση.*

- *Ποικιλομορφία πληροφορίας εντός ενός συνόλου δεδομένων. Εκτός από τις περιπτώσεις όπου έχουμε διαφορετικές πηγές, πολλές φορές ακόμα και δεδομένα που προέρχονται από την ίδια πηγή, σαν μέρος του ίδιου συνόλου, μπορεί να* παρουσιάζουν διαφοροποιήσεις ως προς το μοντέλο που ακολουθούν και την πληροφορία που περιέχουν. Αυτό το χαρακτηριστικό καθιστά πολύ δύσκολη την δημιουργία ερωτημάτων που θα ανταποκρίνονται σε όλες τις διαφοροποιήσεις καθώς και την ανεύρεση της ζητούμενης πληροφορίας. Συστήματα που έχουν ως στόχο την πλήρη εξερεύνηση συνόλων δεδομένων πρέπει να λαμβάνουν υπόψη τους όλα τα πιθανά μοντέλα καθώς και τις πιθανές διαφοροποιήσεις στην μοντελοποίηση της πληροφορίας, όπως αυτές εμφανίζονται στην ιεραρχία των σημασιολογικών επισημάνσεων.

Στην συνέχεια, παρουσιάζονται οι προκλήσεις που αφορούν τις λειτουργικότητες και τα σενάρια χρήσης, δίνοντας έμφαση στις δυσκολίες που προκύπτουν από αυτές για την σχεδίαση ενός συστήματος που να ανταποκρίνεται πλήρως στις ανάγκες των χρηστών. Αυτές είναι:

- Διαδραστική εξερεύνηση και οπτικοποίηση. Το βασικό ζητούμενο των χρηστών είναι να αποκτήσουν πρόσβαση στην πληροφορία που τους ενδιαφέρει με έναν απλό τρόπο που θα τους επιτρέπει όμως να εξερευνούν την πληροφορία σε πολλαπλά αφαιρετικά επίπεδα και μέσω πολλαπλών φίλτρων.

- Απώλεια πληροφορίας. Κομβικό ζητούμενο για εφαρμογές που διαχειρίζονται μεγάλα σύνολα δεδομένων είναι να εξασφαλίζεται η πρόσβαση σε όλη την διαθέσιμη πληροφορία. Δεδομένου του όγκου των δεδομένων, και της δυσκολίας που υπάρχει σχετικά με την διαχείριση του, πάρα πολλές λύσεις ξεκινάμε με την ομαδοποίηση ή την περίληψη των δεδομένων. Τέτοιες λύσεις, ενώ αρχικά είναι πολύ βοηθητικές γιατί επιτρέπουν στον χρήστη να έχει μια πρώτη εικόνα των δεδομένων και του τι περιλαμβάνεται στο σύνολο στην συνέχεια προκαλούν προβλήματα στην βαθιά εξερεύνηση της πληροφορίας, λόγω της πληροφορίας που αποκρύπτεται. Στην προσπάθεια να περιοριστεί αυτό το ζήτημα πολλά συστήματα υιοθετούν την πρακτική των λεπτομερειών με βάση τις επιλογές του χρήστη, δυναμικά αποκρύπτοντας ή παρουσιάζοντας δεδομένα με βάση τις επιλογές εξερεύνησης.

- Προσβασιμότητα. Ένα ακόμα πολύ σημαντικό ζητούμενο για συστήματα που διαχειρίζονται μεγάλα σύνολα δεδομένων είναι το να παρέχουν πρόσβαση στα δεδομένα, πριν και μετά την επεξεργασία, χωρίς να απαιτούν από τον χρήστη να έχει ακριβό εξοπλισμό και τοπικές υπολογιστικές δυνατότητες. Είναι επίσης πολύ σημαντικό το σύστημα να είναι προσβάσιμο από πολλούς χρήστες ταυτόχρονα και να είναι ευέλικτο σε ότι αφορά τις πηγές και τα σύνολα δεδομένων που διαχειρίζεται.

- Υποστήριξη ερωτημάτων. Παρόλο που οι λειτουργικότητες που έχουν συζητηθεί μέχρι τώρα αφορούν χρήστες που δεν έχουν καμία πρότερη γνώση σε ότι αφορά τα μεγάλα σύνολα δεδομένων και τον Σημασιολογικό Ιστό, υπάρχει και μια ακόμα κατηγορία χρηστών που έχουν περισσότερες γνώσεις. Οι χρήστες αυτοί ενδιαφέρονται όχι μόνο για την εξερεύνηση της πληροφορίας όπως αυτή έχει επιλεχθεί μέσα από το σύστημα αλλά και για την δυνατότητα δημιουργίας ερωτημάτων και οπτικοποίησης των αποτελεσμάτων που θα αφορούν συγκεκριμένες λεπτομέρειες του συνόλου των δεδομένων.

Στην διδακτορική αυτή μελέτη, προτείνονται λύσεις για τις προκλήσεις που αναφέρθηκαν παραπάνω ανάλογα με τα σενάρια χρήσης και το προφίλ των ατόμων που *αναμένονται να τα χρησιμοποιήσουν.*

Οπτικοποίηση της πληροφορίας βασισμένη σε χαρακτηριστικά του SPARQL ερωτήματος. *Η πρώτη λύση που προτείνεται, στοχεύει σε έμπειρους χρήστες που είναι τουλάχιστον σε ικανοποιητικό βαθμό εξοικειωμένοι με την γλώσσα ερωτημάτων SPARQL και επιθυμούν να εξερευνήσουν σύνολα δεδομένων που ενημερώνονται συχνά, είναι μεγάλα ή περιέχουν περίπλοκες διασυνδέσεις και προσφέρονται υπό το Πλαίσιο Περιγραφής Πόρων μέσω ενός τελικού σημείου σύνδεσης SPARQL. Ένα χαρακτηριστικό τέτοιο σύνολο δεδομένων είναι το Standard-Thesaurus Wirtschaft (STW) Thesaurus for Economics, ένα λεξικό με αγγλικούς και γερμανικούς όρους οικονομικών το οποίο ενημερώνεται πολύ συχνά, είτε με την ανάπτυξη των ορισμών ήδη διαθέσιμων λέξεων είτε με την προσθήκη νέων όρων.*

Το λεξικό μπορεί να χρησιμοποιηθεί από οικονομολόγους που ψάχνουν τον κατάλληλο όρο να χρησιμοποιήσουν, μεταφραστές που ενδιαφέρονται να μεταφράσουν τεχνικά κείμενα αλλά και ερευνητές γλωσσολόγους που ενδιαφέρονται να κάνουν περίπλοκα ερωτήματα και αναζητήσεις για να δουν τις αλλαγές στην χρήση των όρων ανάλογα με την εποχή ή συσχετίσεις ανάμεσα σε όρους και ερμηνείες.

Για την επίλυση τέτοιων αναγκών χρήσης και εξερεύνησης προτείνεται ένα ολοκληρωμένο σύστημα το οποίο μπορεί να χρησιμοποιηθεί πάνω σε οποιοδήποτε SPARQL σύνολο δεδομένων που είναι διαθέσιμο μέσα από ένα τελικό σημείο σύνδεσης, ανεξαρτήτως των χαρακτηριστικών του. Η προτεινόμενη λύση περιλαμβάνει ένα σύστημα υποστήριξης λήψεων αποφάσεων που συμβάλει στην επιλογή της κατάλληλης οπτικοποίησης για κάθε ερώτημα SPARQL που μπορεί να δημιουργήσει ο χρήστης, βασισμένο σε μια βάση γνώσεων που περιλαμβάνει τα αποτελέσματα μια εκτεταμένης πειραματικής μελέτης κατά την οποία αναλύθηκαν συγκεκριμένα χαρακτηριστικά πολλών SPARQL συνόλων δεδομένων.

Το σύστημα περιλαμβάνει ακόμα μια διασυνδεδεμένη πλατφόρμα και μια διεπαφή χρήστη που επιτρέπει την δημιουργία και την εκτέλεση ερωτημάτων με την χρήση της SPARQL γλώσσας ερωτημάτων, προσφέρει την οπτικοποίηση των αποτελεσμάτων με τον βέλτιστο δυνατό τρόπο και μια σειρά από λειτουργικότητες όπως αναζήτηση με λέξη-κλειδί. Το σύστημα συνοδεύεται ακόμα από μια μελέτη για τα κριτήρια και τις παραμέτρους που συμβάλλουν στην επιλογή της κατάλληλης οπτικοποίησης για κάθε πιθανό ερώτημα.

Για την υποστήριξη ενός τέτοιου συστήματος έχει προταθεί μια αρχιτεκτονική πελάτη-εξυπηρετητή η οποία περιλαμβάνει μια διασυνδεδεμένη πλατφόρμα και ένα έξυπνο σύστημα υποστήριξης λήψεων αποφάσεων. Η πλατφόρμα, που αποτελεί τον πελάτη του αρχιτεκτονικού σχεδιασμού, είναι υπεύθυνη για την αλληλεπίδραση με τον χρήστη, την καταγραφή και απομακρυσμένη εκτέλεση των ερωτημάτων SPARQL καθώς και την εξαγωγή των χαρακτηριστικών για την χρήση του συστήματος υποστήριξης λήψεων αποφάσεων. Ο εξυπηρετητής, χρησιμοποιεί τα χαρακτηριστικά του ερωτήματος για να προτείνει στην πλατφόρμα την κατάλληλη οπτικοποίηση για το συγκεκριμένο ερώτημα με βάση τους κανόνες που έχουν θεσπιστεί από το σύστημα υποστήριξης λήψεων αποφάσεων.

Για την πειραματική μελέτη τα παρακάτω ερωτήματα εκτελέστηκαν σε 55 σύνολα δεδομένων που ήταν διαθέσιμα την περίοδο των δοκιμών. Τα ερωτήματα αυτά είναι:

- *Result size limit. SELECT ?subject ?predicate ?object WHERE {?subject ?predicate ?object}*

- *Number of unique predicates. SELECT (count(distinct ?predicate) as ?count) WHERE {?subject ?predicate ?object}*

- *Number of unique subjects. SELECT (count(distinct ?subject) as ?count) WHERE {?subject ?predicate ?object}*

- *Number of unique objects. SELECT (count(distinct ?object) as ?count) WHERE {?subject ?predicate ?object}*

- *Number of predicates. SELECT (count(?predicate) as ?count) WHERE {?subject ?predicate ?object}*

- *Minimum appearances of predicates. SELECT ?predicate (count(?predicate) as ?count) WHERE {?subject ?predicate ?object} GROUP BY ?predicate ORDER BY ASC(?count) LIMIT 1*

- *Minimum appearances of objects. SELECT ?object (count(?object) as ?count) WHERE {?subject ?predicate ?object} GROUP BY ?object ORDER BY ASC(?count) LIMIT 1*

- *Maximum appearances of predicates. SELECT ?predicate (count(?predicate) as ?count) WHERE {?subject ?predicate ?object} GROUP BY ?predicate ORDER BY DESC(?count) LIMIT 1*

- *Maximum appearances of objects. SELECT ?object (count(?object) as ?count) WHERE {?subject ?predicate ?object} GROUP BY ?object ORDER BY DESC(?count) LIMIT 1*

- *Minimum string length for predicates. SELECT ?predicate (strlen(str(?predicate)) as ?min) WHERE {?subject ?predicate ?object } ORDER BY ASC(?min) LIMIT 1*

- *Minimum string length for subjects. SELECT ?subject (strlen(str(?subject)) as ?min) WHERE {?subject ?predicate ?object } ORDER BY ASC(?min) LIMIT 1*

- *Minimum string length for objects. SELECT ?object (strlen(str(?object)) as ?min) WHERE {?subject ?predicate ?object } ORDER BY ASC(?min) LIMIT 1*

- *Maximum string length for predicates. SELECT ?predicate (strlen(str(?predicate)) as ?max) WHERE {?subject ?predicate ?object } ORDER BY desc(?max) LIMIT 1*

- *Maximum string length for subjects. SELECT ?subject (strlen(str(?subject)) as ?max) WHERE {?subject ?predicate ?object } ORDER BY desc(?max)*

- *Maximum string length for objects. SELECT ?object (strlen(str(?object)) as ?max) WHERE {?subject ?predicate ?object } ORDER BY desc(?max)*

Τα αποτελέσματα από τα παραπάνω ερωτήματα χρησιμοποιήθηκαν από την βάση γνώσεων ως το βασικό μέτρο σύγκρισης για τις παραμέτρους και την προτεινόμενη οπτικοποίηση. Τα πιθανά μήκη ακμών, ο αριθμός ακμών ανά κόμβο αλλά και ο ρυθμός επανεμφάνισης των κόμβων είναι στοιχεία που μπορούν να χρησιμοποιηθούν ώστε να καθοριστούν βασικοί παράμετροι της γραφικής απεικόνισης της πληροφορίας, όπως είναι το μήκος των ακμών, το μέγεθος των κόμβων και το ποσοστό της επιτρεπόμενης επικάλυψης στα στοιχεία του γράφου.

Εξερεύνηση της πληροφορίας. Παρουσιάζεται εδώ μια λύση η οποία στοχεύει στο να βοηθήσει χρήστες που δεν είναι εξοικειωμένοι με τα μεγάλα σύνολα δεδομένων και τον Σημασιολογικό Ιστό στο να εξερευνήσουν σύνολα δεδομένων τα οποία δεν ενημερώνονται συχνά αλλά περιέχουν σημαντικές πληροφορίες που πρέπει να εξερευνηθούν σε βάθος. Για αυτήν την περίπτωση είναι απαραίτητο να δημιουργηθεί μια ολοκληρωμένη πλατφόρμα που να μπορεί να διαχειρίζεται σύνολα δεδομένων ανεξάρτητα από τα χαρακτηριστικά τους, να ανταποκρίνεται το ίδιο αποτελεσματικά σε οποιοδήποτε μέγεθος και αν έχει το σύνολο δεδομένων, να αποθηκεύει την πληροφορία αποτελεσματικά και να προσφέρει σωστή δεικτοδότηση της χωρικής πληροφορίας καθώς και μια διεπαφή χρήστη. Η διεπαφή χρήστη θα πρέπει να προσφέρει πρόσβαση στην πληροφορία για πολλούς χρήστες ταυτόχρονα, σε περισσότερα από ένα σύνολα δεδομένων καθώς και ένα σύνολο από λειτουργικότητες που θα συμβάλουν στην ομαλή εξερεύνηση της πληροφορίας και στην αναζήτηση των δεδομένων που ενδιαφέρουν τον χρήστη.

Η παραπάνω λύση βασίζεται σε μια προ-επεξεργασία των δεδομένων σε τρία στάδια. Το πρώτο στάδιο της τεχνικής επικεντρώνεται στην διαχείριση συνόλων δεδομένων ανεξαρτήτως μεγέθους. Για να το πετύχει αυτό, το αρχικό σύνολο δεδομένων χωρίζεται σε πολλά μικρότερα τμήματα. Ο αριθμός και το μέγεθος αυτών των τμημάτων αποφασίζονται με βάση το μέγεθος, τον αριθμό των οντοτήτων, των αριθμό των σχέσεων καθώς και βασικών χαρακτηριστικών του αρχικού συνόλου.

Τα χαρακτηριστικά που λαμβάνονται υπόψη είναι ο βαθμός συνδεσιμότητας των κόμβων, το μήκος των ετικετών για τους κόμβους και τις ακμές, η παρουσία σημασιολογικής πληροφορίας και η παρουσία μεγάλων κειμένων. Οι ακμές που αφαιρούνται κατά την διάσπαση του συνόλου των δεδομένων σε μικρότερα τμήματα αποθηκεύονται κατάλληλα ώστε να ενωθούν και πάλι με τα τμήματα του συνόλου δεδομένων κατά το τρίτο στάδιο της τεχνικής. Αυτό είναι ιδιαίτερα σημαντικό, δεδομένου ότι κύρια τεχνική απαίτηση για την πλατφόρμα είναι η διαχείριση ολόκληρης της αρχικής πληροφορίας.

Το δεύτερο στάδιο της τεχνικής επικεντρώνεται στην σωστή αναπαράσταση των δεδομένων που περιλαμβάνονται στα τμήματα στα οποία έχει διασπαστεί το αρχικό σύνολο δεδομένων. Κάθε κομμάτι μετατρέπεται σε έναν μικρό ανεξάρτητο γράφο με την χρήστη του αλγορίθμου Scalable Force Directed Placement. Η επιλογή του αλγορίθμου βασίζεται στο γεγονός ότι προσφέρει μεγάλη ελαστικότητα σε ότι αφορά την παραμετροποίηση του αποτελέσματος και υψηλή αποδοτικότητα ακόμα και σε περιπτώσεις όπου τα δεδομένα είναι περίπλοκα. Με βάση αυτό και λαμβάνοντας υπόψη τα χαρακτηριστικά των τμημάτων του αρχικού συνόλου δεδομένων, επιλέγονται οι κατάλληλοι παράμετροι του αλγορίθμου για την δημιουργία του γράφου.

Στόχος είναι να επιτευχθεί η δημιουργία ενός γράφου ο οποίος θα είναι συμπαγής στον κατάλληλο βαθμό ώστε να μην υπάρχουν επικαλύψεις ανάμεσα στους κόμβους αλλά και όλα τα τμήματα να οπτικοποιούνται καταλαμβάνοντας περίπου την ίδια έκταση στον χώρο. Όπως θα δούμε στην συνέχεια, αυτό είναι πολύ σημαντικό για το τρίτο στάδιο της τεχνικής που παρουσιάζεται.

Το τρίτο στάδιο της τεχνικής είναι αφιερωμένο στην δημιουργία ενός ενιαίου *γράφου στον χώρο που θα προσφέρει εξερεύνηση της πληροφορίας, μετατρέποντας κάθε οντότητα του αρχικού συνόλου δεδομένων σε έναν κόμβο ενός ενιαίου γράφου με συγκεκριμένες συντεταγμένες στον χώρο. Είναι πολύ σημαντικό οι υπογράφοι να τοποθετηθούν στον χώρο με τέτοιο τρόπο ώστε οι συνδέσεις μεταξύ τους, αυτές που αφαιρέθηκαν και αποθηκεύτηκαν ξεχωριστά στο πρώτο στάδιο της τεχνικής, να εισαχθούν στον ενιαίο γράφο με τέτοιο τρόπο ώστε να έχουν το ελάχιστον δυνατό μήκος.*

Για τον λόγο αυτόν, χρησιμοποιείται ένας ευριστικός αλγόριθμος, βασισμένος σε μια συνάρτηση κόστους πάνω στο μήκος των εξωτερικών ακμών που συνδέουν τους υπογράφους, που έχει σχεδιαστεί κατάλληλα ώστε να τοποθετηθούν οι υπογράφοι στον χώρο με την βέλτιστη δυνατή κατανομή, ελαχιστοποιώντας το συνολικό μήκος των εξωτερικών ακμών.

Η ομογενοποιημένη πληροφορία αποθηκεύεται σε μια βάση δεδομένων κατάλληλη για την αποθήκευση χωρικής πληροφορίας, την Geomesa, και δεικτοδοτείται κατάλληλα με ένα XZ-index. Η πληροφορία είναι πλέον διαθέσιμη για την χρήση της από άλλα υπολογιστικά συστήματα και την διεπαφή χρήστη που έχει δημιουργηθεί για την υποστήριξη την εξερεύνησης της πληροφορίας σαν μέρος του συστήματος. Στα πλαίσια της προτεινόμενης λύσης, η διεπαφή του χρήστη προσφέρει μια σειρά από λειτουργικότητες που περιλαμβάνουν την εποπτεία ολόκληρου του οπτικοποιημένου συνόλου δεδομένων, την διαδραστική οπτικοποίηση της πληροφορίας και την παρουσίαση της μέσα από πολλαπλά επίπεδα εστίασης και φιλτραρίσματος, την αναζήτηση με λέξη κλειδιά, την εξερεύνηση με βάση μονοπάτια ενδιαφέροντος και την απομόνωση τμήματος του γράφου για την καλλίτερη εξερεύνηση του.

Σημασιολογική Εξερεύνηση της πληροφορίας. *Η διαθεσιμότητα των μεγάλων συνόλων δεδομένων έχει οδηγήσει στην ανάπτυξη πολλών συστημάτων τεχνικής νοημοσύνης και μηχανικής μάθησης που στοχεύουν στο να εξάγουν γνώση από αυτά τα δεδομένα. Αρχικά, ο διαθέσιμος όγκος των δεδομένων θεωρήθηκε αρκετός για την ανάπτυξη αξιόπιστων συστημάτων που θα μπορούσαν να εξάγουν με αντικειμενικό τρόπο την γνώση. Βασική πρόκληση της τεχνικής νοημοσύνης, όμως, είναι το γεγονός ότι δεν εφαρμόζει κανέναν έλεγχο στα δεδομένα που διαχειρίζεται, δεν μπορεί να αναγνωρίσει αν κάποια είναι μη έγκυρα, περιέχουν υποκειμενική πληροφορία ή πληροφορία που δεν είναι θεματικά ισορροπημένη, να ανταποκριθεί σε ημιδομημένα ή ελλιπή σύνολα δεδομένων. Αυτό είχε σαν αποτέλεσμα την δημιουργία πολλών συστημάτων που δεν ανταποκρίθηκαν στους σκοπούς για τους οποίους προορίζονταν.*

Το πιο γνωστό ίσως τέτοιο παράδειγμα είναι η αποτυχία του πρώτου Watson για Oncology. Το 2013, η IBM συνεργάστηκε με το The University of Texas MD Anderson Cancer Center για να αναπτύξει ένα σύστημα που θα χρησιμοποιούσε την πλούσια βάση δεδομένων του κέντρου καρκίνου για να μπορέσει να προσφέρει επιπλέον γνώσεις και καθοδήγηση για την αντιμετώπιση περιστατικών καρκίνων με πιο αποτελεσματικό τρόπο. Τα αρχικά αποτελέσματα έδειξαν ότι το σύστημα αποτύγχανε συνεχώς να προτείνει την σωστή θεραπεία για τις περιπτώσεις καρκίνων που κλήθηκε να αντιμετωπίσει. Αποδείχθηκε τελικά ότι το βασικό πρόβλημα του συστήματος ήταν το σύνολο δεδομένων που χρησιμοποιήθηκε για την εκπαίδευση του συστήματος και αυτό γιατί πολλές από τις πληροφορίες είχαν αποκρυφθεί ή τροποποιηθεί για την προστασία των ασθενών.

Τα chatbots, συστήματα εκπαιδευμένα να συνομιλούν ανταλλάσσοντας γραπτά μη-

νύματα με χρήστες, προσφέροντας υπηρεσίες τεχνικής υποστήριξης ή υπηρεσίες προς καταναλωτές και πελάτες, δεν είναι άτρωτα σε τέτοια προβλήματα. Το πιο γνωστό σύστημα που απέτυχε παταγωδώς και πάρα πολύ γρήγορα ήταν ο Tay, ένα chatbot που σχεδιάστηκε από την Microsoft, για να μπορεί να απαντάει αυτόματα σε μηνύματα στην πλατφόρμα κοινωνικής δικτύωσης Twitter. Ο στόχος του συστήματος ήταν να παράγει μηνύματα που θα ταιριάζουν στα πρότυπα των μηνυμάτων που ανταλλάσσονται σε πλατφόρμες κοινωνικών δικτύων, να είναι δηλαδή γραμμένα σε απλή, καθημερινή γλώσσα, να περιέχουν συντμήσεις, αστεία και ιδιωματισμούς.

Το σύστημα φιλοδοξούσε να αποκτήσει γνώσεις για την επικοινωνία μέσω γραπτού λόγου και να βελτιώνει την απόδοση του και το πόσο ρεαλιστικά ήταν τα μηνύματα του μέσα από την επανεκπαίδευση με μηνύματα που λάμβανε. Οι χρήστες της πλατφόρμα κοινωνικής δικτύωσης Twitter όμως αποφάσισαν να παίξουν με το σύστημα και άρχισαν να του στέλνουν ρατσιστικά και σεξιστικά μηνύματα και πολύ σύντομα ο Tay απαντούσε με παρόμοιο προσβλητικό τρόπο, χρησιμοποιώντας λέξεις και φράσεις που περιείχαν αυτά τα μηνύματα, αναδεικνύοντας την έλλειψη ελέγχων για τα δεδομένα που χρησιμοποιήθηκαν για την επανεκπαίδευση.

Τα παραπάνω παραδείγματα κάνουν σαφές ότι παρόλο που ο διαθέσιμος όγκος των μεγάλων όγκων δεδομένων είναι πολύ σημαντικός για τα συστήματα τεχνητής νοημοσύνης και μηχανικής μάθησης, η πηγή και η ποιότητα των δεδομένων μπορούν να επηρεάσουν καθοριστικά την απόδοση τους και θα πρέπει να λαμβάνονται σημαντικά υπόψη στον αρχιτεκτονικό σχεδιασμό τέτοιων συστημάτων. Στα πλαίσια των συστημάτων συζήτησης και διαλόγων υπάρχουν πολλά διαθέσιμα σύνολα δεδομένων με διαλόγους και συζητήσεις που έχουν καταγραφεί άμεσα ή έμμεσα από ανθρώπινες επαφές. Τα περισσότερα από αυτά τα σύνολα δεδομένων ακολουθούν το μοντέλο ζεύγους ερώτηση-απάντηση και χρησιμοποιούν μεγάλες πηγές δεδομένων όπως είναι η Wikipedia και οι Yahoo Answers. Ένα σημαντικό ποσοστό από τα υπόλοιπα σύνολα δεδομένων έχουν συλλεχθεί κατά την διάρκεια υποστήριξης καταναλωτών μέσα από ομαδικά ή προσωπικά συστήματα ανταλλαγής μηνυμάτων. Υπάρχουν ακόμα κάποια σύνολα δεδομένων που περιέχουν διαλόγους από ταινίες, διαλόγους ανάμεσα σε συστήματα διαλόγων τεχνητής νοημοσύνης ή δημόσιες συζητήσεις.

Αυτά τα δεδομένα κατασκευάζονται με βάση την υπάρχουσα πληροφορία και τα δεδομένα που είναι διαθέσιμα και μπορεί να περιέχουν πολλά διαφορετικά θέματα, χωρίς να δίνεται καμία σημασία στην αξία που μπορεί να αποκομίσει κανείς από την επεξεργασία αυτών των δεδομένων. Η διαδικασία και ο σκοπός για τον οποίο δημιουργούνται αυτά τα σύνολα δεδομένων έχουν σαν αποτέλεσμα να υπάρχουν μια σειρά προκλήσεων στην προσπάθεια χρήσης αυτών των δεδομένων για την δημιουργία συστημάτων δημιουργίας διαλόγων. Αυτές οι προκλήσεις είναι:

- ***Μη-ισορροπημένη και υποκειμενική πληροφορία.*** *Τα γλωσσικά ιδιώματα και το λεξιλόγιο που χρησιμοποιείται περισσότερο ανά περίσταση μπορεί πολύ εύκολα να εισάγει μια υποκειμενικότητα στο σύστημα διαλόγων. Χαρακτηριστικό παράδειγμα θα ήταν η εκπαίδευση με την χρήση ενός συνόλου δεδομένων που έχει προκύψει από ταινίες δράσης ενός συστήματος για την τεχνική υποστήριξη καταναλωτών. Είναι προφανές ότι το σύστημα δεν θα έχει ούτε το απαιτούμενο λεξιλόγιο ούτε τις κατάλληλες γνώσεις συντακτικού και γραμματικής που χρειάζονται σε αυτήν την περίπτωση χρήσης. Ακόμα όμως και αν ένα σύνολο δεδομένων είναι συλλεγμένο με τέτοιον τρόπο ώστε να περιέχει αντιπροσωπευτική πληροφορία μπορεί και πάλι να είναι υποκειμενικό ή να περιέχει στερεότυπα. Για παράδειγμα, ένα σύνολο δεδομένων που έχει συλλεχθεί από* ένα τηλεφωνικό κέντρο που διαχειρίζεται παράπονα χρηστών για ελλαττωματικά προϊόντα είναι αναμενόμενο να είναι αρνητικά φορτισμένο συναισθηματικά και να περιέχει λεξιλόγιο που σχετίζεται με απογοήτευση και εκνευρισμό.

- ***Ελλιπή σύνολα δεδομένων.*** *Όταν τα σύνολα δεδομένων δημιουργούνται με αυτόματο τρόπο είναι πιθανό να περιέχουν διαλόγους που δεν έχουν ολοκληρωθεί ή έχουν διακοπεί λόγω τεχνικών προβλημάτων. Τέτοιες περιπτώσεις δεν είναι εύκολο να βρεθούν με αυτόματο τρόπο οδηγώντας σε σύνολα δεδομένων που είναι ελλιπή ή και λανθασμένα.*

- ***Θεματική ποικιλία.*** *Ένα διευρυμένο λεξιλόγιο είναι το βασικό ζητούμενο για πολλά συστήματα δημιουργίας διαλόγων και υπάρχουν πολλοί σχετικοί αλγόριθμοι που εστιάζουν στο να συγκεντρώσουν όσο το δυνατών περισσότερες λέξεις. Μια τέτοια τεχνική μπορεί να προσφέρει πολλά σε συστήματα γενικού σκοπού, όπως ένα σύστημα που απαντάει μηνύματα κοινωνικών δικτύων, αφού είναι πολύ πιθανό να έρθουν αντιμέτωπα με πολλά και διαφορετικά θέματα συζήτησης. Κάτι τέτοιο όμως πιθανότατα θα μπερδέψει ένα σύστημα ειδικού σκοπού που πρέπει να βοηθήσει σε συγκεκριμένη δουλειά. Για παράδειγμα ένα σύστημα τεχνικής υποστήριξης θα πρέπει να αναγνωρίζει την λέξη ποντίκι σαν μια συσκευή εισόδου που χρησιμοποιείται στους ηλεκτρονικούς υπολογιστές και όχι σαν ένα μικρό ζώο της κλάσης των θηλαστικών και της τάξης των τρωκτικών, για να είναι αποδοτικό.*

- ***Γλωσσική ιδιομορφία.*** *Εξίσου σημαντικό με το να δημιουργηθεί ένα σύνολο δεδομένων που να είναι σημασιολογικά σωστό για το θέμα για το οποίο θα χρησιμοποιηθεί, είναι και το να περιέχει πληροφορίες εκφρασμένες με τον κατάλληλο τρόπο για την δουλειά στην οποία θα χρησιμοποιηθεί. Ένα σύστημα διαλόγων που έχει σχεδιαστεί για να απαντάει σε μηνύματα σε πλατφόρμες κοινωνικής δικτύωσης θα πρέπει να χρησιμοποιεί απλή και καθημερινή γλώσσα, αστεία και απλές εκφράσεις. Αντίστοιχα, ένα σύστημα που έχει σχεδιαστεί για να προσφέρει τεχνική υποστήριξη σε χρήστες θα πρέπει να χρησιμοποιεί σωστούς γραμματικούς κανόνες, σωστό συντακτικό και να προσφέρει σωστή καθοδήγηση και οδηγίες ώστε να βοηθήσει τον χρήστη στην επίλυση του προβλήματος.*

Για την επίλυση των προκλήσεων που περιεγράφηκαν παραπάνω, και έχοντας ως στόχο την αξιοποίηση των συνόλων δεδομένων που είναι διαθέσιμα με τέτοιο τρόπο που να εξαλείφονται τα μη-αξιοποιήσιμα και υποκειμενικά δεδομένα, αναπτύχθηκαν τεχνικές σημασιολογικής ανάλυσης των δεδομένων. Η σημασιολογική ανάλυση των δεδομένων βασίζεται κυρίως στις σημασιολογικές σχέσεις ανάμεσα σε όρους, όπως αυτές καθορίζονται από την γλωσσολογία. Μεγάλη ερευνητική προσπάθεια έχει αφιερωθεί στις σημασιολογικές σχέσεις, το πως αυτές καθορίζονται και το ποια θα είναι η ακριβής συσχέτιση ανάμεσα στους όρους που θα τις υπακούσουν. Οι βασικές σημασιολογικές σχέσεις είναι:

- *Συνώνυμα. Λέξεις που έχουν το ίδιο ακριβώς ή πολύ κοντινό νόημα, όπως είναι το αυτοκίνητο και το όχημα.*

- *Αντώνυμα. Λέξεις που έχουν ακριβώς το αντίθετο νόημα, όπως είναι το ζεστό και το κρύο.*

- Εξειδικεύσεις. Δύο λέξεις συνδέονται με μία σχέση εξειδίκευσης όταν ο ένας όρος είναι πιο συγκεκριμένος από τον άλλον για παράδειγμα ανάμεσα στις λέξεις χρώμα και κόκκινο. Σε αυτήν την περίπτωση η λέξη κόκκινο είναι εξειδίκευση της λέξης χρώμα.

- Γενικεύσεις. Δύο λέξεις συνδέονται με μία σχέση γενίκευσης όταν ο ένας όρος είναι πιο γενικός από τον άλλον για παράδειγμα ανάμεσα στις λέξεις πιρούνι και σερβίτσιο. Εδώ το πιρούνι είναι μια εξειδίκευση της λέξης σερβίτσιο ενώ το σερβίτσιο είναι μία γενίκευση της λέξης πιρούνι.

- Μέρος ενός συνόλου. Είναι η σχέση που συνδέει ένα τμήμα με το σύνολο στο οποίο αντιστοιχεί, είναι για παράδειγμα η σχέση που συνδέει το δέντρο με το δάσος.

Οι παραπάνω σημασιολογικοί κανόνες καθορίζουν την σημασιολογική ομοιότητα και την σημασιολογική σχετικότητα ανάμεσα σε όρους και φράσεις. Πολύ συχνά οι δύο όροι χρησιμοποιούνται σαν να είναι συνώνυμοι, κυρίως γιατί η σημασιολογική σχετικότητα θεωρείται μια πιο απλοποιημένη και γενικευμένη έκφραση της ομοιότητας. Στην πραγματικότητα, σύμφωνα με τους αυστηρούς ορισμούς των δύο όρων η πιο σημαντική τους διαφοροποίηση είναι οι σημασιολογικές σχέσεις που μπορούν να ληφθούν υπόψη και στις δύο περιπτώσεις και το βάρος που αυτές έχουν στον καθορισμό της σημασιολογικής ομοιότητας ή σχετικότητας.

Πιο συγκεκριμένα, η σημασιολογική ομοιότητα είναι μια μετρική που εξετάζει το κοινό σημασιολογικό περιεχόμενο, την ομοιότητα των λέξεων που χρησιμοποιούνται με βάση συγκεκριμένες βάσεις γνώσεως, και βασίζεται στα συνώνυμα, στις εξειδικεύσεις και στις γενικεύσεις. Η σημασιολογική σχετικότητα είναι μια πιο διευρυμένη μετρική που εξετάζει την σημασιολογική εγγύτητα, βασισμένη σε οποιαδήποτε σημασιολογική σχέση ανάμεσα σε όρους, ακόμα και όταν όροι είναι αντώνυμα ή μέρη ενός συνόλου.

Για την βελτίωση της ποιότητας των διαθέσιμων συνόλων δεδομένων που χρησιμοποιούνται για την εκπαίδευση συστημάτων διαλόγων, προτείνουμε μια τεχνική που ορίζει μια αυστηρή ροή δεδομένων και εξασφαλίζει ότι τα σύνολα δεδομένων που δίνονται ως είσοδο επεξεργάζονται με τον καλύτερο δυνατό τρόπο τόσο με βάση τον σημασιολογικό προσανατολισμό του συστήματος όσο και με βάση την περίπτωση χρήσης. Κάθε ζεύγος ερώτηση-απάντηση του συνόλου εισόδου λαμβάνει έναν βαθμό σημασιολογικής ομοιότητας και ένα βαθμό σημασιολογικής σχετικότητας, οι οποίοι είναι ανεξάρτητοι μεταξύ τους.

Οι δύο αυτοί βαθμοί συνενώνονται μέσα από μια συνάρτηση που χρησιμοποιεί βάρη, τα οποία καθορίζονται με βάση την πηγή του συνόλου δεδομένων από το οποίο προέρχεται το ζεύγος ερώτηση-απάντηση καθώς και το ύφος της γλώσσας που το σύστημα θα πρέπει να χρησιμοποιήσει. Το τελικό αποτέλεσμα είναι ένα ποσοστό καταλληλότητας για το κάθε ζεύγος ερώτηση-απάντηση, το οποίο μπορεί να χρησιμοποιηθεί για να κατατάξει την πληροφορία και να επιτρέψει τον περιορισμό του συνόλου δεδομένων με βάση το κατάλληλο κατώφλι.

Μεταβλητές. Η προτεινόμενη τεχνική χρειάζεται τρεις βασικές μεταβλητές εισόδου. Η πρώτη μεταβλητή αφορά το θέμα ενδιαφέροντος για το αυτόματο σύστημα διαλόγων και εκφράζεται με ένα σύνολο λέξεων-κλειδιών που το αντιπροσωπεύουν με τον καλύτερο δυνατό τρόπο. Το σύνολο των λέξεων-κλειδιών μπορεί να περιέχει μέχρι και πέντε λέξεις, τόσο για να εξασφαλιστεί η σημασιολογική συνοχή των λέξεων και κατά επέκταση του παραγόμενου συνόλου δεδομένων όσο και για να διευκολύνει την σημασιολογική ανάλυση με τον περιορισμό πιθανών μην συσχετιζόμενων σημασιολογικών σχέσεων.

Η δεύτερη παράμετρος που χρειάζεται η τεχνική είναι ο προσδιορισμός του σεναρίου χρήσης για το αυτόματο σύστημα διαλόγων. Αυτή η πληροφορία είναι απαραίτητη για τον προσδιορισμό των βαρών στην συνάρτηση που ενώνει τους βαθμούς σημασιολογικής σχετικότητας και ομοιότητας στο τελικό ποσοστό καταλληλότητας για το κάθε ζεύγος ερώτηση-απάντηση ώστε να δοθεί προτεραιότητα στην πληροφορία που έρχεται από πιο κατάλληλες πηγές. Η συγκεκριμένη πληροφορία δίνεται με δύο τρόπους, πρώτα επιλέγεται το ύφος της γλώσσας που θα χρησιμοποιηθεί, ο χρήστης μπορεί να επιλέξει με βάση το αν θα είναι απλή, επίσημη ή ημι-επίσημη. Στην συνέχεια το βασικό σενάριο χρήσης για το σύστημα αυτόματων διαλόγων επιλέγεται.

Οι διαθέσιμες επιλογές αντιστοιχίζονται πλήρως στις διαθέσιμες κατηγορίες και πηγές συνόλων δεδομένων που είναι διαθέσιμα και περιλαμβάνουν κατηγορίες όπως υποστήριξη χρηστών και απάντηση σε μηνύματα που δημοσιεύονται σε πλατφόρμες κοινωνικών δικτύων. Αυτό επιτρέπει την ακόμα καλύτερη προσαρμογή των βαρών στην συνάρτηση που ενώνει τους βαθμούς σημασιολογικής σχετικότητας και ομοιότητας στο τελικό ποσοστό καταλληλότητας ώστε να δίνεται καλύτερη βαθμολογία σε πηγές που είναι πολύ κοντινές με τα σενάρια χρήσης.

Η τρίτη παράμετρος είναι τα σύνολα δεδομένων που θα χρησιμοποιηθούν για την εξαγωγή του ζητούμενο συνόλου δεδομένων. Δεδομένης της ποικιλομορφίας των επιλογών, είναι αναμενόμενο να υπάρχουν σύνολα δεδομένων που να καλύπτουν όλες τις ανάγκες σε ποιότητα, προέλευση, θέμα και ύφος γλώσσας κάνοντας τα είτε ιδανικά είτε τελείως ακατάλληλα για την χρήση τους ανάλογα το σενάριο χρήσης. Επιπλέον, οι χρήστες μπορούν να επιλέξουν σαν είσοδο και ένα σύνολο δεδομένων που έχουν οι ίδιοι με την προϋπόθεση ότι ακολουθεί το μοντέλο ζεύγους ερώτηση-απάντηση.

Για τους σκοπούς της σημασιολογικής ανάλυσης η WordNet λεξιλογική βάση δεδομένων χρησιμοποιείται καθώς προσφέρει ομάδες από συνώνυμες λέξεις καθώς και συσχετίσεις μεταξύ τους βασισμένες στις σημασιολογικές σχέσεις που συζητήθηκαν. Η ροή δεδομένων που έχει επιλεχθεί για την επεξεργασία των συνόλων δεδομένων εισόδου περιγράφεται εδώ. Η τεχνική διαχειρίζεται ένα-ένα κάθε ζεύγος ερώτηση-απάντηση που υπάρχει στα σύνολα δεδομένων εισόδου. Το πρώτο βήμα είναι ο υπολογισμός του βαθμού σημασιολογικής ομοιότητας ανάμεσα σε κάθε ένα ζεύγος ερώτηση-απάντηση και τις λέξεις-κλειδιά που έχουν επιλεγεί από τον χρήστη. Ένας αλγόριθμος σχεδιασμένος να λαμβάνει υπόψη του μόνο τις κατάλληλες σημασιολογικές σχέσεις υπολογίζει την σημασιολογική σχετικότητα και επιστρέφει ένα κανονικοποιημένο βαθμό.

Στην συνέχεια υπολογίζεται η σημασιολογική σχετικότητας ανάμεσα σε κάθε ένα ζεύγος ερώτηση-απάντηση και τις λέξεις-κλειδιά που έχουν επιλεγεί από τον χρήστη. Σε αυτήν την περίπτωση ο αλγόριθμος λαμβάνει υπόψη του όλες τις σημασιολογικές σχέσεις που υπάρχουν ανάμεσα στα δύο εξεταζόμενα σύνολα λέξεων και επιστρέφει και πάλι ένα κανονικοποιημένο βαθμό. Τέλος, η συνάρτηση βαρών ενώνει τους δύο βαθμούς, λαμβάνοντας υπόψη την πηγή του συνόλου δεδομένων και το σενάριο χρήσης, δίνοντας το τελικό ποσοστό καταλληλότητας για το ζεύγος ερώτηση-απάντηση.

Μετά την επεξεργασία όλων των ζευγών ερώτηση-απάντηση των συνόλων δεδομένων εισόδου, τα δεδομένα κατατάσσονται με βάση το τελικό ποσοστό καταλληλότητας. Σαν ένα πρώτο βήμα, για να διευκολυνθεί η μετέπειτα επεξεργασία της πληροφορίας και να μειωθεί το μέγεθος του αποτελέσματος, ζεύγη ερώτηση-απάντηση με τελικό ποσοστό καταλληλότητας μικρότερο από 50% απορρίπτονται σαν ακατάλληλα *να χρησιμοποιηθούν για αυτό το θέμα και το σενάριο χρήσης. Τα δεδομένα γίνονται διαθέσιμα σε ένα αρχείο κειμένου που ακολουθεί το Comma-Separated Values μοντέλο σε τριπλέτες πληροφορίας. Κάθε τριπλέτα περιλαμβάνει το ζεύγος ερώτηση-απάντηση και το τελικό ποσοστό καταλληλότητας, έχει δηλαδή την δομή (ερώτηση, απάντηση, τελικό ποσοστό καταλληλότητας).*

Ποιοτικός Έλεγχος δεδομένων. *Η καταγραφή των αλλαγών σε υδάτινα περιβάλλοντα είναι μια πάρα πολύ δύσκολη, ακριβή και χρονοβόρα διαδικασία που βασίζεται στην χρήση συγκεκριμένου εξοπλισμού. Στην προσπάθεια να αλλάξει αυτό, έχουν δημιουργηθεί νέα καινοτόμα εργαλεία που επιτρέπουν στους πολίτες να συμμετέχουν στην παρακολούθηση του περιβάλλοντος χρησιμοποιώντας έξυπνες εφαρμογές και αισθητήρες.*

Τα δεδομένα που συλλέγονται από τους πολίτες με αυτόν τον τρόπο θα χρησιμοποιηθούν τόσο από επιστήμονες όπως μηχανικοί και υδρολόγοι όσο και από τις τοπικές Αρχές προκειμένου να αποκτήσουν μια ακριβέστερη εικόνα των αλλαγών στο τοπικό περιβάλλον και των αναγκαίων παρεμβάσεων.

Οι εφαρμογές έχουν ως στόχο να αναδείξουν τις τεράστιες δυνατότητες παρατήρησης και παρακολούθησης του περιβάλλοντος από τους πολίτες, που μπορούν να καταγράψουν τις μεταβολές στην κάλυψη και χρήση της γης αλλά και μετρήσεις από αισθητήρες όπως η υγρασία του εδάφους και η θερμοκρασία του αέρα. Τα δεδομένα που συλλέγονται από τους εθελοντές θα βοηθήσουν στη βελτίωση της ακρίβειας των υφιστάμενων χαρτών κάλυψης και χρήσης γης και στην αποτελεσματικότερη διαχείριση των πλημμυρικών φαινομένων.

Η εφαρμογή "Scent Explore" είναι ένα παιχνίδι επαυξημένης πραγματικότητας. Οι χρήστες κερδίζουν πόντους βρίσκοντας τους διασκεδαστικούς χαρακτήρες της εφαρμογής, που είναι κρυμμένοι στο τοπικό τους περιβάλλον. Οι χαρακτήρες είναι τοποθετημένοι σε συγκεκριμένα σημεία περιβαλλοντικού ενδιαφέροντος, με σκοπό τη συλλογή εικόνων και βίντεο, που απεικονίζουν αλλαγές στην κάλυψη και τη χρήση γης, καθώς και πληροφοριών για τις διάφορες παραμέτρους το ποταμού, όπως τη στάθμη και την ταχύτητα του νερού.

Η εφαρμογή "Scent Measure" επιτρέπει στους πολίτες να συμβάλουν στην παρατήρηση των αλλαγών στις συνθήκες του εδάφους. Η εφαρμογή χρησιμοποιεί φορητούς αισθητήρες για τη λήψη μετρήσεων σχετικών με την υγρασία του εδάφους και τη θερμοκρασίας του αέρα. Οι χρήστες απλά τοποθετούν τον έξυπνο αισθητήρα μέσα στο έδαφος και συλλέγουν τις μετρήσεις στο κινητό τους.

Όπως είναι αναμενόμενο σε κάθε μεγάλο σύνολο δεδομένων έτσι και εδώ η ποιότητα και η αξιοπιστία των μετρήσεων που έχουν συλλεχθεί είναι αμφισβητούμενη. Οι εφαρμογές σχεδιάστηκαν ώστε να καθοδηγήσουν τους εθελοντές όσο το δυνατόν καλύτερα στην συλλογή δεδομένων που θα είναι αξιόπιστα και θα ανταποκρίνονται στις πραγματικές συνθήκες του περιβάλλοντος. Πολλοί από τους εθελοντές όμως δεν είχαν τις κατάλληλες τεχνολογικές γνώσεις για την χρήση όλων των λειτουργικοτήτων των εφαρμογών ενώ πολλές φορές οι συνθήκες στο πεδίο ήταν τέτοιες που δεν επέτρεπαν την συλλογή των δεδομένων με τον κατάλληλο τρόπο.

Την ίδια στιγμή η βελτίωση των υδρολογικών μοντέλων στις περιοχές που μελετήθηκαν, απαιτούσαν ακριβή και αξιόπιστα δεδομένα που θα συνέβαλαν στην βελτίωση της απόδοσης τους. Για τον λόγο αυτόν αναπτύχθηκε ένας μηχανισμός ελέγχου της ποιότητας των δεδομένων που βασίστηκε σε μια σειρά από κανόνες και πρακτικούς περιορισμούς.

Για τα δεδομένα κάλυψης γης οι εικόνες που έχουν συλλεχθεί είναι ξανά δια-

θέσιμες στους εθελοντές όπου μπορούν να προσθέσουν επιπλέον επισημάνσεις ή να προτείνουν αλλαγές και βελτιώσεις σε ήδη υπάρχουσες. Σε κάθε περίπτωση, για να θεωρηθεί μια επισήμανση έγκυρη πρέπει να γίνει αποδεκτή από την πλειοψηφία των εθελοντών που έχουν κληθεί να χαρακτηρίσουν την εικόνα ως προς την κάλυψη γης. Σε περιπτώσεις όπου επιλογές δεν έχουν ξεκάθαρη επικρατούσα βάση, ειδικό βάρος δίνεται στις επιλογές του εθελοντή που κατέγραψε την εικόνα στο πεδίο, δεδομένου ότι έχει καλύτερη εικόνα της κατάστασης και γνώση στοιχείων που πιθανώς δεν καταγράφονται πλήρως ή ξεκάθαρα στην φωτογραφία.

Σε ότι αφορά την στάθμη του νερού και την ταχύτητα ροής των υγρών σωμάτων, αυτές είναι πληροφορίες που προέρχονται από εικόνες και βίντεο των εθελοντών που έχουν υποστεί επεξεργασία από κατάλληλους αλγορίθμους, αυτό όμως δεν σημαίνει ότι έχουν μη έγκυρες μετρήσεις είτε λόγω λανθασμένης καταγραφής είτε λόγω κάποιου προβλήματος με την ακρίβεια των αλγορίθμων. Για να βεβαιωθεί η εγκυρότητα των μετρήσεων που συλλέγονται με αυτόν τον τρόπο έχει προσδιοριστεί μια συγκεκριμένη μεθοδολογία.

Το πρώτο βήμα της μεθοδολογίας είναι η χωρική και χρονική ομαδοποίηση των μετρήσεων. Γνωρίζοντας από τον τρόπο διεξαγωγής των μετρήσεων ότι οι εθελοντές κινούνται κατά ομάδες, συλλέγουν δεδομένα ταυτόχρονα και πάντα στα προκαθορισμένα σημεία ενδιαφέροντος μπορούν πολύ εύκολα να δημιουργηθούν οι κατάλληλες ομάδες μετρήσεων που αφορούν την ίδια καταγραφή.

Στο επόμενο βήμα της τεχνικής ελέγχεται η εσωτερική διαφορά σε κάθε ομάδα, που αναφέρεται και ως ο έλεγχος σίγμα. Ο έλεγχος αυτός εξετάζει αν τα δεδομένα που ανήκουν σε μια ομάδα ακολουθούν την κανονική κατανομή, η μέση τιμή και η αναμενόμενη απόκλιση υπολογίζονται προσεγγιστικά από τα γενικά χαρακτηριστικά του δείγματος. Μετρήσεις που απέχουν πολύ από την αναμενόμενη κανονική κατανομή απομακρύνονται από το σύνολο των δεδομένων.

Σε ότι αφορά τις μετρήσεις υγρασίας εδάφους και θερμοκρασίας αέρα που προκύπτουν από τους φορητούς αισθητήρες η πληροφορία που τις συνοδεύει, συντεταγμένες, χρόνος καταγραφής, μοναδική ταυτότητα μέτρησης, μοναδική ταυτότητα αισθητήρα και μοναδική ταυτότητα χρήστη χρησιμοποιείται με βάση μια σειρά κανόνων για την ποιοτική αξιολόγηση τους. Ο πρώτος κανόνας αφορά χωρικά απομονωμένες μετρήσεις. Γνωρίζοντας, όπως αναφέρθηκε ήδη, από τον τρόπο διεξαγωγής των μετρήσεων ότι οι εθελοντές κινούνται κατά ομάδες των τριών τουλάχιστον ατόμων έχουν οριστεί τα δέκα μέτρα σαν το όριο που μια μέτρηση μπορεί να απέχει από όλες τις άλλες καθώς μια τέτοια μέτρηση είναι είτε μη αξιόπιστη είτε πολύ μακριά από το σημείο ενδιαφέροντος.

Ο επόμενος κανόνας αφορά μετρήσεις που έρχονται αποκλειστικά και μόνο από έναν εθελοντή. Αν για μια περιοχή, μόνο ένας εθελοντής παρέχει μετρήσεις, κάτι που δεν είναι εφικτό με βάση τον τρόπο διεξαγωγής των μετρήσεων, το πιο πιθανό είναι να υπάρχει κάποιο πρόβλημα με τις συντεταγμένες ή την αρχικοποίηση του αισθητήρα και οι μετρήσεις που δίνονται να μην είναι αξιόπιστες.

Τέλος, ελέγχεται η ακρίβεια της μέτρησης των συντεταγμένων όπως αυτή καταγράφεται από τον ίδιο τον αισθητήρα. Πολλές από τις μετρήσεις έγιναν σε απομακρυσμένες περιοχές ή μέσα στις όχθες ποταμών όπου υπήρχε έντονη κάλυψη από πυκνή βλάστηση με αποτέλεσμα πολλές μετρήσεις να έχουν πολύ μεγάλα σφάλματα στην καταγραφή των συντεταγμένων. Μετρήσεις με απόκλιση μεγαλύτερη των είκοσι μέτρων αφαιρέθηκαν από το σύνολο δεδομένων.

Επιπρόσθετα για τις μετρήσεις θερμοκρασίας αέρα εξετάστηκε και ο έλεγχος δι-

αστήματος. Σαν αξιόπιστο διάστημα για την θερμοκρασία καθορίστηκε μια απόκλιση δέκα βαθμών Celsius από την μέση θερμοκρασία που είχαν καταγράψει οι εθελοντές μέσα στην ημέρα. Αντίστοιχα για τις μετρήσεις υγρασίας εδάφους εξετάστηκε και ο έλεγχος δέλτα. Η ανάλυση των μετρήσεων που αφορούσαν την υγρασία εδάφους έδειξε ότι υπήρχαν πάρα πολλές μετρήσεις με τιμή 0%.

Εκτενής πειραματική μελέτη σε εργαστηριακές συνθήκες έδειξαν ότι ο αισθητήρας μπορεί να καταγράψει έγκυρη μέτρηση 0% όταν τοποθετείται σε ξερό έδαφος που δεν έχει καθόλου ρίζες ζωντανών φυτών. Υπάρχει όμως και η περίπτωση η τιμή 0% να καταγραφεί γιατί ο αισθητήρας δεν χρησιμοποιήθηκε σύμφωνα με τις οδηγίες, η πρώτη μέτρηση καταγράφηκε πιο γρήγορα από τα δεκαπέντε δευτερόλεπτα που χρειάζονται για να καταγραφεί η πρώτη μέτρηση.

Για να διαφοροποιηθούν οι δύο περιπτώσεις μηδενικών μετρήσεων, εξετάζεται η χρονοσειρά με τις μετρήσεις που κατέγραψε ο συγκεκριμένος χρήστης με τον συγκεκριμένο αισθητήρα αμέσως μετά την μηδενική μέτρηση. Αν μέσα στα επόμενα δεκαπέντε δευτερόλεπτα υπάρχει μέτρηση με τιμή μεγαλύτερη 0% τότε η μέτρηση καταγράφεται σαν λάθος χρήση του αισθητήρα και αφαιρείται από το σύνολο δεδομένων.

List of Tables

Abbreviations

2D	:	*Two dimensional*
3D	:	*Three dimensional*
ACID	:	*Atomicity, Consistency, Isolation & Durability*
AI	:	*Artificial Intelligence s*
API	:	*Application programming interface*
BSP	:	*Binary Space Partitioning*
CLC	:	*CORINE Land Cover*
CORINE	:	*Coordination of Information on the Environment*
CPU	:	*Central Processing Unit*
CQL	:	*Common Query Language*
CRUD	:	*Create, Read, Update and Delete)*
CS	:	*Citizen Science*
CSV	:	*Comma-Separated Values*
DB	:	*Data Base*
DBMS	:	*Database Management System*
DSS	:	*Decision Support System*
FoI	:	*Feature of Interest*
GAP	:	*Grid Arrangement Problem*
GB	:	*Gigabyte*
GHz	:	*Gigahertz*
GPS	:	*Global Positioning System*
GSM	:	*Global System for Mobile Communications*
HHCode	:	*Helical Hyperspatial Code*
HRTA	:	*ellenic RescueTeam of Attica*
HTML	:	*HyperText Markup Language*
HTTP	:	*Hypertext Transfer Protocol*
ID	:	*Identification*
IoT	:	*Internet of Things*
ISO	:	*International Organization for Standardization*
IVLG	:	*Interactive Visualization of Very Large Graphs*
JSON	:	*JavaScript Object Notation*
KB	:	*Knowledge Database*
LC/LU	:	*Land Cover/ Land Use*
LDVM	:	*Linked Data Visualization Model*
LIWC	:	*Linguistic Inquiry and Word Count*
LOD	:	*Linked Open Data*
LSTM	:	*Long Short-Term Memory*
MB	:	*Megabyte*
Mbps	:	*Megabits per second*
MMU	:	*Minimum Mapping Unit*

NLP	:	*Natural Language Processing*
NP-hard	:	*Non-deterministic Polynomial-time Hard*
OASIS	:	*Organization for the Advancement of Structured Information Standards*
OGC	:	*Open Geospatial Consortium*
OSM	:	*Open Street Map*
OWL	:	*Web Ontology Language*
PDOP	:	*Position Dilution of Precision*
PoI	:	*Point of Interest*
POMS	:	*Profile of Mood States*
px	:	*Pixel*
QA	:	*Question-Answer*
RAM	:	*Random Access Memory*
RDBMS	:	*Relational Database Management System*
RDF	:	*Resource Description Framework*
REST	:	*REpresentational State Transfer*
RFID	:	*Radio-frequency identification*
SFDP	:	*Scalable Force Directed Placement*
SGD	:	*Stochastic Gradient Descent*
SNAP	:	*Grouping Nodes on Attributes and Pairwise Relationships*
SPARQL	:	*SPARQL Protocol and RDF Query Language*
SQL	:	*Structured Query Language*
STW	:	*Standard-Thesaurus Wirtschaft*
SVM	:	*Support Vector Machine*
UI	:	*User Interface*
UNESCO	:	*United Nations Educational, Scientific and Cultural Organization*
UN	:	*United Nations*
URI	:	*Uniform Resource Identifier*
URL	:	*Uniform Resource Locator*
VQB	:	*Visual Query Builder*
W3C	:	*World Wide Web Consortium*
WLMT	:	*Water Level Measurement Tool*
WVCT	:	*Water Velocity Calculation Tool*
WWW	:	*World Wide Web*
XML	:	*Extensible Markup Language*

Chapter 1

Introduction

It was only twenty years ago when Tim Berners-Lee initiated the idea about the Semantic Web [29]. Its growth and expansion the following years was exponential as more and more data providers understood the importance of a common way to share information for their development and sustainability. A slow start, hindered by the need to manually annotate semantic data, was soon followed by a rapid acceleration when the process was automated. Encyclopedias, medical and scientific databases, large-scale projects such as Bio2RDF, the British Museum, the BBC Programmes and Music and the most prominent project Wikidata, the large knowledge graph of Wikipedia, were soon part of the Semantic Web and offered under the RDF format.

RDF is a data interchange model that aims to support the merging of information from heterogeneous sources with different schemas as well as the unobstructed update of schemas as needed without requiring any modification to the consumers of the information. RDF takes advantage of the structure of the Web and utilizes URIs to indicate relationships between entities. This data model provides the flexibility to merge structured and semi-structured schemas and share them in a uniform way. What makes the RDF model so widely used is that it is flexible and can be used to model information from heterogeneous sources. This is also what makes it a challenge to explore and understand, as data expressed in RDF is typically stored in large interconnected databases, without a homogeneous schema.

Upon the increase of the volume of the information available on the Web, the need for a uniform way that facilitates the accessibility, the discovery and the understanding of the available information became prominent.

Since being released as an official W3C recommendation in early 2008, SPARQL has evolved into the major query language for the Semantic Web and the RDF model. It has become the main standard for querying semantic data stores and is a key technology of the open data movement. SPARQL is supported by nearly all modern RDF based storage systems and is widely used in enterprise and public web contexts.

There are two main challenges regarding the use of the SPARQL query language for the exploration of information. To begin with, novice users of SPARQL can easily perform simple and basic queries but require extensive training to utilize all the exploration capabilities of the language due to the wide range of functionalities. This is often deterring or disengaging for users that are not very familiar with the Semantic Web. In addition, the provided data rarely comply with strict models or have specific structure. Most users however are familiar with strict data models and rela-

tional databases as well as results that are complete and have specific structure. The diversity and flexibility or the SPARQL results is often hindering to the compilation of meaningful queries and the understanding of the information.

A SPARQL endpoint can be strictly defined as a conformant SPARQL protocol service according to the SPARQL Protocol for RDF specification. A SPARQL endpoint enables users, either humans or machines, to query a knowledge base using the SPARQL language. Results are typically returned in machine-processable formats, mainly SPARQL Query Results XML or JSON Format. Informally, a SPARQL endpoint is mostly conceived as a machine-friendly interface over a knowledge database. Accessing the information, forming the proper queries, understanding and representing the retrieved information is the responsibility of the user of the endpoint and not of the provider. As the number of the available endpoints increases, information for many different research areas is available, which is of interest to a wider audience, especially scientists and researchers, with limited knowledge of the Semantic Web.

To address the needs of such users a few endpoints were updated to offer access to the information through search engines and present the information in structured ways. These efforts however are limited to a very small number of the available endpoints and offer few, if any, exploration and visualization functionalities. The structured formats used, such as lists and tables, present the information in a human-readable format but without showing relations and connections between entities.

Recently, the wide adoption of the Linked Open Data initiative has made available large linked datasets [185] that have the potential to offer invaluable knowledge. Accessing, evaluating and understanding these datasets as published, though, requires extensive training and experience in the field of the Semantic Web, making these valuable sources of information inaccessible to a wider audience.

In the recent years there have been many efforts to find optimal techniques to process such datasets, extract the contained information and presented it to the user in meaningful ways. These techniques, however, impose a series of restrictions to the volume and characteristics of the input dataset, limiting their applicability to specific use cases.

A common requirement for existing systems is the availability of complete or partial semantic annotations for the input dataset. These systems, such as [7, 59], exploit the semantic annotations, to significantly reduce the volume of data that are required to handle. Based on the hierarchy defined by the annotations, they provide the functionality for link navigation and representation of semantic data resources and their properties. These systems work only on datasets with a limited amount of semantic classes and ignore any non annotated information. This way not only they limit the number of possible navigation paths that are required to handle but also limit the overall volume of the accessible data. In addition, any information of the dataset that is not semantically annotated is never presented to the user.

Other systems extract information contained in linked, semantically annotated, datasets and aggregate it, based on its type as provided by the annotation, and presented through generic visualization options such as diagrams and charts. Some systems such as [15, 41, 171] only handle predefined types of information and restrict the users to these, while others [269, 321] provide to the user dynamic options that can be configured. Another approach, as presented in [301], is based on a heuristic analysis of the structure of the input data and a comprehensive visualization model to *enable the semi-automatic binding between data and visualization parameters. Such systems provide to the user only informative overviews of the dataset. They fail, however, to present the complete information and result in loss of information as data that do not comply with the identified semantic annotations are discarded.*

Some systems target specific types of data or vocabularies. Recognizing the importance of the spatial information some systems focus on visualizing and exploring geo-spatial data. Such systems, as [182], provide a faceted browsing tool that enables RDF datasets to be visualized on an OSM or Google Map while others, as [288], provide an exploration and visualization tool for SPARQL accessible data, offering faceted filtering functionalities.

Other systems, such as [133], target multidimensional linked data modelled with the Data Cube vocabulary. They provide faceted browsers for exploring these data using different types of visualizations and charts. In these cases, there are some strict limitations regarding the input datasets along with the possibility of loss of information depending on its completeness.

A large number of systems visualize linked datasets adopting a graph-based approach [202]. Most systems, limit the displayed information by enforcing a path-based navigation of the data to the user. Such examples, like [130], are web-based tools that offer interactive discovery and visualization of relationships between selected linked data resources. There are also some exploratory tools, such as [126, 45] that allow users to browse linked data using interactive graph navigation. Starting from a given URI, the user can explore linked data by following the links. Other systems, such as [339], employ a space-optimized visualization algorithm in order to display additional resources with respect to the user navigation choices, or propose, as in [73], a clustered RDF graph visualization in which input information is merged to graph nodes. In all these cases the user is given access only to some of the input information and very limited exploitation options.

Aiming to eliminate the requirement for semantic annotations, the dataset structure was exploited as a mean to filter the dataset and limit the volume of the handled data. The most well-known and popular technique is based on the hierarchical model. Such techniques, as the ones presented in [3, 13, 155], present the input dataset either filtered or aggregated, in order to limit the volume that they are required to handle.

Although the hierarchical approaches provide interactive visualizations with low memory requirements, they have two main drawbacks. To begin with, their applicability is heavily based on the particular characteristics of the input dataset as they can be applied only on acyclic datasets that conform to a hierarchical data model. Moreover, the hierarchical approaches result in loss of any information not complying with the initially identified model, making accessible by the user only part of the information.

Next, tools were developed that explore the information retrieved from SPARQL endpoints in structured ways. Many web-based tools [139, 324] were developed for displaying, accessing, filtering and exploring query results as obtained by SPARQL endpoints. The structured formats used in such tools, such as lists and tables, present the information in a human-readable format but without showing relations and connections between entities.

The exponential increase of the number and size of the available linked datasets as well as the diversity of their characteristics [297], increases their usefulness, im-

portance and appeal to a wider audience. Only specific use cases and datasets that comply with strict limitations and characteristics can be explored through the systems presented above. There are many interesting, important and useful categories of datasets that are generated in a random way that do not comply with any of these requirements. These categories include road maps, communication networks, biological structures, financial and blockchain transactions where the datasets are complicated, highly connected, contain critical information that should be available complete and accessible by many users with different degrees of knowledge for the Semantic Web and also different usability purposes. This makes of utmost importance the identification of a generic technique that will be scalable and independent of any specific characteristic of the input dataset that can be used for their exploration.

Following the Linked Open Data initiative, Big Data characterization for published datasets is increasingly popular and it is used in many research domains and scientific works. Carefully examining its usage, however, reveals it is, more often than not, used with continuously changing meanings. The different definitions available for the term [325, 66, 67] are inconsistent and vary depending on the time they were written and the scientific field that they are referring to. What all the definitions have in common are some key characteristics, including that they are large, complex and unprocessed datasets, that cannot be processed by traditional applications but can offer knowledge and value if properly analyzed.

Initially, there was a controversy regarding the volume of the dataset and what should be considered large or difficult to process. This is mainly due to the fact that the application-specific capacity of the machines to compute information per capita has roughly doubled every 14 months, whereas the world's storage capacity per capita required roughly 40 months to double during the last decades [137]. The exponential growth of the data production, the diversity of the data sources, along with the improvement of the computational capabilities of the hardware made the quantification of the term insignificant but added multiple dimensions to the problem.

To this end, a dataset is now characterized as Big Data based on its dimensions or the Vs that it complies to [136]. Starting from the basic three Vs [219], volume, velocity and variety, the definition was soon updated to four Vs [234] that include also veracity, which is the more commonly accepted definition. Next, value was added as the 5^{th} *V [90, 70]. Value is the first indirect characteristic of Big Data, as it is referring to the desired outcome of their processing [312]. The five dimensions of the Big Data can be defined as:*

- *Volume. The amount of data is an important aspect of the Big Data given that the goal is to process high volumes of low-density, unstructured data. Here, the high volume is not defined by the state-of-the-art capabilities but it is specific for each consumer of the information and it is defined by the quantity of the generated data, the storage capabilities and their ability to process and analyze them. For some applications, this might be as low as tens of terabytes of data while for others it may easily reach hundreds of petabytes.*

- *Variety. It refers to the type and nature of the data. Big Data distance themselves from structured data schemas that fit neatly in relational databases. Nowadays, the majority of the raw data are available in unstructured and semi-structured data types including video, audio or metadata, such as click-streams* *on a web page. These data introduce the overhead of additional pre-processing to derive the included information.*

- *Velocity. In this context, the velocity is referring to two different aspects. On the one hand, with the term velocity we are addressing the speed at which the data are generated and/or updated. This identifies the rate that the system should achieve for the storage of the information in order to avoid any loss. Big Data are expected to be produced continually. On the other hand, Big Data analysis is expected to achieve near real-time results so the velocity is also referring to the frequency of handling, processing, and publishing the results of the data manipulation.*

- *Veracity. It merges together the unreliability of the data sources and the reliability of the data forming a dataset [137]. Veracity is discussing the danger of a data source to change the quality or the content of the data provided [105].*

- *Value. The sheer volume of the data available combined with the diversity of the data sources and the lack of official requirements often produce datasets that are of low quality that fail to deliver what was expected. To this end, a dataset is considered as Big Data only when its processing and analysis produces value for a user. The importance of this aspect is highlighted by the most popular definition for the Big Data, the one presented by McKinsey's Business Technology Office [199], which was the first to associate the utilization of big data sources to the creation of added value to an economic field.*

The availability of the Big Data has led to the development of many artificial intelligence and machine learning methods that extract valuable information from the datasets. Initially, the volume of the available data was deemed enough to develop reliable systems that could extract the available knowledge. Artificial intelligence, however, cannot identify invalid data, outliers, bias contributions and unbalanced information or cope with incomplete and unstructured datasets. As a result, many erogenous systems were developed [299, 111].

Aiming to support Artificial intelligence solutions, to take advantage of the available datasets in ways that are protected by invalid data and bias contributions, semantic analysis techniques are employed to improve the content of the datasets. Semantic analysis methods are mainly based on the semantic relationships between terms as defined in linguistics. A lot of effort is dedicated in defining these semantic relationships in ways that will showcase both the similarity and the relatedness of words. The formal definitions of the main semantic relationships that are going to be used in the upcoming algorithms, are presented here:

- *Synonymy. Two terms are characterized as synonymous when they have exactly or nearly the same meaning, such as the terms car and automobile.*

- *Antonymy. Two terms are characterized as antonymous when they represent the complete opposite from one other, such as hot and cold.*

- *Hyponymy. It is used to show a relationship of specification, such as the relationship between the world color and red. In this case, red is a hyponym of color.*

- *Hypernymy. It is used to show a relationship of generalization, such as the relationship between the world fork and the general term cutlery. Here, cutlery is a hypernym fork.*

- *Meronymy. It is a relationship connecting a part to its whole. As an example, a tree is a meronym of a forest.*

The terms of semantic similarity and relatedness are often used interchangeably, mainly because semantic relatedness is often seen as a casual, generalized similarity. Their main difference, however, is the semantic relationships used in their calculation. The definition of the terms is presented here:

Semantic Similarity. *A metric used to evaluate two documents, terms or sets of terms with regard to their common semantic content, the likeness of their meaning based on knowledge sources. Two entities are considered semantically similar when they are associated with what is commonly refer to as 'is a' semantic relationships which are synonymy, hyponymy and hypernymy.*

Semantic Relatedness. *A broader metric, that extents the semantic similarity, that evaluates two documents, terms or sets of terms with regard to their semantic closeness. Two entities are considered semantically related when they are associated with any linguistic relationship, including meronymy or antonymy.*

Adopting a different approach to the acquisition of useful and meaningful datasets Kevin Ashton, co-founder and executive director of the Auto-ID Labs [179], presented the idea of the Internet of Things (IoT) at a presentation that aimed to introduce the Radio-frequency identification (RFID) technology to the production chain of Procter & Gamble (P&G) in 1999 [149]. The proposed innovation was the usage of the RFID technology [97], specifically of electromagnetic fields, to automatically identify and track tags attached to products. An RFID tag is a low-cost, tiny radio transponder, which is a receiver and transmitter. This tag can be triggered by an electromagnetic interrogation pulse from a RFID reader device, prompting it to transmit pre-defined information, a unique identifying number. The main idea is that this tag can be read in an automated way at specific places during the supply chain and be used as a tracking method for the produced goods as well as a way to analyse the production lines, the time between stations, identify bottlenecks and improve the overall production flow.

The key idea that the term IoT aimed to introduce was the disengagement of a human from the production of useful, meaningful and exploitable information. Until the early 2000's the majority of the data that were available via the internet were collected by human beings through repetitive and time consuming tasks, such as capturing an image, typing text and filling forms or scanning barcodes to track products. The data collected through such tasks, however, were often incomplete, of low quality and inconsistent, mainly due to the lack of enthusiasm of the humans towards the tedious tasks that they had to perform. The IoT envisioned a network of sensors that could collect the needed information in an automated way reducing the possibility of errors, and ensuring high quality and consistent data aimed to tackle these challenges [14].

After the first usage of the term, new concepts were added to the definition, to imply a shared understanding of the situation and the environment among humans and appliances, and the term was also associated with the design of overall archi-

tectures that could efficiently support such concepts with both software and hardware solutions, as well as a pervasive communication network that allows the collection, transmission and processing of the information[16, 192]. Last but not least, the definition of the Internet of Things included the need for efficient and effective data analysis applications that ensures an autonomous and creative behavior from the system. All these additions, aimed for the design of smartly connected network of sensors that will allow content-aware data analysis. The definition has recently been more inclusive regarding the applications for which an IoT system can be of value. Besides, the traditional industrial systems, smart-home products, healthcare appliances and fitness trackers, utilities, as well as smart transport systems, have been introduced to the IoT ecosystem [295].

All these new applications, as well as the recent evolution in technology, has made necessary a more flexible definition regarding what can be identified as 'Things'. These changes, however, have not changed the main goal of the IoT systems, which can be seen as making any hardware platform sense, understand and interpret the current status of its environment without any input or support from human users. The architectural design of the IoT systems has evolved into a network of interconnected objects that are no longer tasked with passively collecting information from their environment but are entrusted to analyse the situation and command and control other objects accordingly [210].

Important role to the development of the IoT systems has the improvement of the wireless network technologies that can be used by the objects to connect. In addition to RFID, Bluetooth, WiFi and GSM networks can support the data exchange between objects. As a result, IoT has the needed means to step out of its infancy and transform the current static Internet into a fully integrated Internet of Things [43]. The Internet revolution has also led to the interconnection between humans at an impressive scale and between humans and machines faster than anyone thought possible. The next step is to achieve an interconnection between machines similar to the one between humans aiming for a fully connected smart environment [115].

It was in 2011 that the number of objected connected to the IoT overtook the number of people living on the planet. Nearly 22 billion IoT devices were deployed worldwide by the end of 2018, and a further 17 billion will be added by 2025, according to the current estimations and growth. While enterprise IoT has been the major driver in recent years, current projections suggest that the connected home will shortly overtake initially computing and then enterprise IoT [204].

Chapter 2

Problem Definition

As discussed above, the exploration and navigation of very large linked datasets can be associated with different data sources, data formats and data models as well as many use cases and usage scenarios. There are some key challenges that affect all the potential expressions of the problem as they are related either with the nature and characteristics of the datasets or with the basic needs of the users to understand and explore the information.

Intuitively, linked datasets should be made available through interactive systems that enable their visual exploration through multiple abstraction and filtering levels and criteria as well as SPARQL queries. Such systems, however, have to tackle significant challenges related to the characteristics of the datasets, especially as the size of the input dataset increases. These are:

Scalability. *The system should handle large datasets with millions of elements, along with datasets that are incomplete, have no semantic annotations or don't follow a specific data model or hierarchy given that the published datasets do not always comply with such requirements.*

Accessibility. *The system should provide access to many datasets and the visualizations should be available for multiple users. The system should be available through commodity hardware and not dependent of expensive infrastructure.*

Querying support. *RDF data, available through SPARQL endpoints, are fully accessible only for experts in the Semantic Web who are accustomed to forming complex SPARQL queries. Even then, navigating and exploring an unfamiliar SPARQL endpoint by hand can be quite laborious. For novice users only superficial exploration is possible without further support. Systems should provide the users with the needed support to explore and query the information, either by offering filtering and aggregation functionalities with any querying needs or by offering support to the compilation of the queries.*

Content presentation. *RDF is designed to facilitate machine interoperability and does not define a visual presentation model since human readability is not one of its stated goals. Presenting content intended for machines in a human understandable way is very challenging. Most systems provide the information in tables. Such representations are intuitively close to what the novice users, familiar with the relational schemas, expect. They are not, however, representative of the nature of the RDF model, the graphical representation of information with connections between entities. Efforts to provide the information through graph-like visualizations* *pose many challenges. Most approaches limit the usage to specific RDF vocabularies, SPARQL query types, domain-specific analysis or already defined attributes.*

Schema identification. *The underlying data schema is not always available, or easily extracted. As a result, filtering and further querying is not always obvious or intuitive, based on the user expertise and the complexity of the schema next steps for the retrieval of the information may not be straightforward. Some systems aim to extract the underlying schema, they are not, however, always successful as they fail for unstructured or overly complex schemas and only offer an overall estimation regarding the underlying schema.*

Unbalanced & Bias information. *Data bias occurs when the distribution of the training data do not reflect the actual environment that the machine learning model will be used in [283]. In the case of Big Data, the multiple sources as well as different data models used to create the datasets create many consistency issues and make their usage challenging.*

Incomplete datasets lacking annotations. *Many datasets are created in a fully-automated way, without any human overview, or from merging sources with different parameters. This has a results many datasets to be incomplete or lacking important information.*

In addition to the data-related challenges, the system should address important challenges that are associated with the use cases, the user requirements and the content of the dataset. These are:

Interactive visualization. *The data should be visualized in a way that will support the exploration of the information. The data objects should be visualized independently of one another allowing the user to modify the visualization as needed.*

No information loss. *The complete dataset should be available to the user. Aggregating, filtering or grouping the information before presenting it may result in the omission of information critical to the understanding of the dataset.*

Data exploration. *The system should ensure that the user can access and explore the information in an interactive way by providing an ample set of exploration functionalities. These functionalities, such as keyword search, path exploration, filtering and aggregation, should be available in a user-friendly way meaning that they do not require knowledge of querying or programming languages from the user.*

Based on the identified challenges, some use cases where it is crucial for them to be addressed are presented here. Real world examples are presented along with corresponding datasets. For these use cases, in the following chapters of this document appropriate solutions will be presented.

Semantic community exploration. *Visualizing and exploring RDF datasets is a computationally intensive and complex task that is hindered by the volume of the datasets, their different characteristics, including the lack of schemas and structure, as well as the different needs of the users of the information. While in most cases the specific details and relationships of an entity are crucial to the understanding of information, exploring billions of elements to locate it can be disengaging and unrealistic. In cases such as road maps, communication networks, biological structures and blockchain transactions where the datasets are of complicated relational nature, highly connected, contain critical information that should be fully explorable and ac-*

cessible by many users with different analysis needs, the widely accepted solution is to present the information through summarization layers.

An indicative and very interesting dataset that falls into this category is the UN Comtrade repository [313]. It is a free, open repository which provides detailed global trade data, including official international trade statistics and relevant analytical tables. It includes data from 216 countries, in a span of 23 years, concerning the trade of 261 commodities. The data are updated on a monthly basis. Data scientists interested in this repository need access to a visualized overview of semantically meaningful communities. Furthermore, they need to be able to explore the visualized overview, in order to understand connections and patterns in the data, as well as drill-in details of specific entities within the communities.

To begin with, the datasets provided by each country are not identical, they may differ in the volume and format of the provided information as well as the language and semantic annotations used. So here it is very important to have a system that can present the information without any restrictions of the characteristics of the input dataset. In addition, the repository is constantly expanding to include additional data so the system should be able to scale without any limit to the size of the input dataset.

For example, such an overview of this dataset may consist of communities with countries that exclusively export or import commodities, countries that have significantly unbalanced or balanced trade, flows of goods based on geographical and economic factors and potential traffic inconsistencies e.g., countries that import and export the same commodities.

SPARQL endpoint exploration. *One of the most characteristic examples of information spaces that are challenging to explore and visualize is dictionary-like datasets. The information contained in them is incomplete, highly connected due to aliases and synonyms, domain-specific or represented in an unstructured way depending on the source of the information. Such datasets are also very valuable as they are utilized by universities, research institutes, public institutions and companies for knowledge organization. Also the format and characteristics of these dataset are important for research in information science as well as in the area of Linked Data and Semantic Web technologies.*

Such an example of SPARQL endpoint that is very challenging to explore is the Standard-Thesaurus Wirtschaft (STW) Thesaurus for Economics [166]. This thesaurus provides 6.000 subject headings in two languages, English and German and it is considered the world's most comprehensive bilingual thesaurus for representing and searching for economics-related content. It utilizes more than 20,000 synonyms to cover not only all economics-related subject areas but also many related subject fields. The STW is published and continuously further developed by the Leibniz-Informationszentrum Wirtschaft [333], the German National Library of Economics, according to the latest changes in the economic terminology.

Due to the importance of the information that it represents the endpoint is of interest to a wide range of users with different needs. Economists and users less familiar with the Semantic Web focus on locating terms of interest, study their relationships with other terms and vocabularies and find the translation of the terms between the two languages. Exploration systems should provide simple access to the information through keyword search, support the execution of exploratory queries through a *user-friendly interface that requires from the user minimum input or knowledge of the Semantic Web, allows the retrieval of information related to one specific term based on its relationships with others and support the representation of the information in an intuitive way as graph.*

Companies and institutions are more interested in exploring in-depth the dataset, discovering statistics regarding connections between synonyms from different subject fields and retrieving answers to complex queries. The users with experience of the Semantic Web should be able to write their own queries and get the proper visualizations.

Big data visualization. *One of the most characteristic examples of scientific data analysis that requires a system that conforms to all the identified challenges, is the study and analysis of biological structures. Structural biology is a branch of molecular biology concerned with the molecular structure of biological macromolecules such as proteins, made up of amino acids, how they acquire the structures they have, and how alterations in their structures affect their function. This subject is of great interest to biologists because macromolecules carry out most of the functions of cells, and it is only by coiling into specific structures that they are able to perform these functions. The Protein Data Bank [1] is an archive with information about the shapes of proteins, nucleic acids, and complex assemblies. Most of the datasets published there follow the Systems Biology Markup Language [273] aiming to ensure model interoperability and semantically correct information. As an example we take the human cap-dependent 48S pre-initiation complex [143] which represents 47 unique protein chains with 116774 atom count.*

Access to the complete information. Scientist aiming to study and analyse this protein complex need to have access to the complete information in a way that will respect the initial spatial relations, without causing any alterations to connections between molecules. Systems that use aggregation or summarization techniques cannot be used for this analysis as they alter the initial connections between atoms. Techniques that exclude part of the input dataset if it does not follow pre-defined data formats cannot be used either, as protein chains are not expected to fully comply with specific data formats and atoms that deviate from them contain important information which should be available to the user. Systems that filter the input dataset based on specific characteristics cannot support the exploration of protein complexes as their analysis is mostly based on low level connections.

Combination of multiple datasets. In addition, scientists need to study the complete biological structures, so the system should be able to combine multiple protein chains and complexes. As an example, the human cap-dependent 48S pre-initiation protein complex has a relatively small number of atoms as it contains only 47 protein chains. Such complexes, however, are combined or expanded with additional protein chains for analysis purposes. The system should be able to incorporate any additional information and offer to the user exploration and filtering services without any limit to the size of the input dataset. Such dynamic changes in the volume and content of the input dataset is a challenge for all of the available systems. These systems are based on complex pre-processing techniques that aim to identify specific data models that conform with the input dataset or perform a semantic analysis and categorization. Due to their complexity, such techniques fail to combine more than one dataset or adapt to a large and complex one.

Multiple filtering criteria. Users interested in such complex datasets also have high demands regarding different and customizable filtering functions that can be used simultaneously. Popular queries during such data analysis include: "isolate specific molecules", "isolate one or more types of connections based on their semantic annotations", "identify the most/least connected molecules", "locate connections between two molecules", "find if two or more molecules are independent to one other" and "identify specific connections in parts of the biological structure that is responsible for a specific function".

The system should offer such exploration queries though a user-friendly interface, without asking the user to write queries over the data. Depending on the way the information is processed, however, systems may not be able to offer such exploration queries. Systems that aggregate the input dataset cannot offer details for one molecule. Systems that present data based on hierarchy levels or through semantic criteria cannot present the connections between two or more molecules or isolate types of connections that are of interest.

Path exploration. Finally, given the importance of the connections between molecules it is crucial to identify all the neighbors of a molecule or follow paths of interest. All of the available systems, allow either the exploration of the outgoing neighbors of a node or the exploration of nodes along the hierarchy levels of the data model. Therefore, the system does not allow the user to intuitively explore the information associated with a node. For datasets such as the human cap-dependent 48S pre-initiation protein complex, users are very interested in locating all the neighbors of a molecule, either incoming or outgoing, as these connections are valuable to the protein structure analysis.

Targeted semantic analysis. *The availability of the Big Data has led to the development of many artificial intelligence and machine learning methods that extract valuable information from the datasets. Initially, the volume of the available data was deemed enough to develop reliable systems that could extract the available knowledge. Artificial intelligence, however, cannot identify invalid data, outliers, bias contributions and unbalanced information or cope with incomplete and unstructured datasets. As a result, many erogenous systems were developed [299, 29].*

The most well-known example of such a failure is probably the first release of the Watson for Oncology [262]. In 2013, IBM worked with The University of Texas MD Anderson Cancer Center to develop a system that was supposed to utilize the cancer center's rich patient and research databases to provide additional knowledge and insights. The initial results showed that the system failed to suggest the proper treatments. The source of the problem was identified as the dataset used to train the software, a small number of hypothetical cancer patients rather than real patient data.

Chatbots are not immune to such issues, as Microsoft's effort to develop Tay, a chatbot that could automatically reply to Twitter messages and engage in casual and playful conversation [253], proved. The chatbot was supposed to support the research around the conversational understanding and self-improve with time. The users of the Twitter though started sending racist and misogynistic messages and soon Tay was responding in similar insulting ways, using content from these messages, proving that the data cleaning and filtering process for the re-training was not robust.

These examples indicate that while the Volume of the Big Data is very important *for applications in the field of Artificial Intelligence and Machine Learning, the Variety and Veracity affect data quality is such degree that should not be disregarded in the architectural design of such systems [174, 296, 124].*

In the context of the conversational interactions there are many datasets available with dialogues and conversations collected directly or indirectly from human interactions [226]. Most of these datasets are formed using the question-answer pattern and utilize the knowledge base of existing repositories such as Wikipedia and Yahoo Answers. Other datasets focus on customer support interactions through chats and forums. There are also datasets that include dialogues, extracted from movie scripts, chats between bots or conversations held in public forums.

These datasets are formed based on the available information and contain data from many different sources that cover many topics and conversation domains. Limited care is given to the Value of the context of these datasets, with efforts focusing mainly on Volume. Due to their creation process, the main challenges associated with the utilization of these datasets for the training of conversational agents are:

Unbalanced & Bias information. *In the context of conversational datasets bias can very easily materialize. For example, a model trained over dialogues taken from action movies and used as a technical support automated system, will have a great challenge understanding the vocabulary needed and will formulate responses using syntax and grammar not as expected. Even if a dataset is carefully designed to be representative, it can still suffer from prejudicial or stereotyping bias, which is not easy to detect or understand. For example, a conversational dataset extracted from a call center that is responsible for complains regarding defective products will be biased with negative emotions and a vocabulary mostly associated with frustration and disappointment.*

Incomplete datasets lacking annotations. *Many question-answer pairs included in datasets created in an automated way may be incorrect or improperly formatted. An indicative example comes from datasets that are using real customer support interactions. Phone calls and chats may be interrupted or left incomplete, incidents not easily identified with an automated way, leading to incomplete information. In addition, labelling dialogues with highly domain and task specific annotations is a labor intensive and time consuming process, limiting the number of available annotated conversational datasets and the topics that they address.*

Topic diversity. *Expanding the vocabulary recognized and used by conversational agents is one of the main goals of many relevant algorithms. Such approach may be beneficial for general purpose virtual agents that may encounter any discussion topic, but serve only to confuse conversational agents created for specific tasks. Encountering words used in different context and associated with diverse terminology may create connections that will distract the agent from the expected discussion topic.*

Language diversity. *Equally important with creating a dataset that it is semantically correct for the topic that it will be used, it is to have a dataset containing information that corresponds to the task that wants to accomplish. A chatbot created to respond to Twitter messages, should use every-day language and simple, witty and funny expressions. A conversational agent designed to provide technical support must give clear instructions, specific guidelines and use proper grammar and syntax rules as mistakes can hinder the understanding of the information.*

Spatial data exploration. *As the number of the available IoT systems increases, so do the opportunities for data integration and analysis. A representative example showing that integration between different IoT systems could be beneficiary, is the investigation of a possible gas leakage in a dense residential area.*

Starting from a phone call of a citizen suspecting that there is a gas leakage in a specific area, the operator can at once access the IoT system of the gas company to verify if there is a possible leakage and in which part of the network. In the ideal case that all the IoT systems are interconnected, after confirming that there is a possible leakage and identifying the specific building block where the incident takes place, the operator would be able to access all the gas sensors available in the residences of the building block, better locating the area where the leakage may be, triggering the fire alarms of the buildings as needed to ensure the fastest evacuation of the buildings.

In addition, the operator can be given access to thermal and visual cameras, to spot people in immediate danger, requiring assistance, to evaluate and guide the firefighters and other first responders to the specific building/apartment, as needed based on the overview of the situation. Furthermore, even in the case where there is no indication of a leakage at the gas company's distribution network, the operator will be able to validate that using the IoT systems of the building of the area, it can be identified if there is a leakage in a specific building or apartment or any other reason for concern, and take the necessary measures.

Such level of integration requires access to many IoT systems based on multiple criteria related to the three-dimensional space. Proper spatial annotation of the information and efficient spatial indexing are crucial for the realization of such scenario.

Chapter 3

Related Work

3.1 Semantic Exploration

3.1.1 Semantic browsers

In order to support users with semantic data utilization and analysis, semantic browsers are used [59, 7]. They provide functionalities for link navigation and representation of semantic resources and their properties. These techniques aim to address the needs for interactive visualization and data exploration and are efficient for small and fully semantically annotated datasets. They fail to handle non-annotated datasets or scale for larger ones as their processing capabilities are relying on the user-provided hardware.

Semantic browsers are used to access fully semantically annotated datasets that are created from homogeneous sources, such as encyclopedias and dictionaries. The Tabulator [28] project is an attempt to demonstrate and utilize the power of linked RDF data with a user-friendly Semantic Web browser that is able to recognize and follow RDF links to other RDF resources based on the user's exploration and analysis. It is a generic browser for linked data on the web without the expectation of providing as intuitive an interface as a domain-specific application but aiming to provide the sort of common user interface tools used in such applications, and to allow domain-specific functionality to be loaded transparently from the web and be instantly applicable to any new domain of information.

The Linked Data Visualization Model (LDVM) [41, 42] allows to dynamically connect data with visualizations. In order to achieve such flexibility and a high degree of automation, the LDVM is based on a visualization workflow incorporating analytical extraction and visual abstraction steps. Each of the visualization workflow steps comprises a number of transformation operators, which can be defined in a declarative way. As a result, the LDVM balances between flexibility of visualization options and efficiency of implementation or configuration. This has been expanded [133] to support the visualization of data provided using the RDF Data Cube Vocabulary.

RelFinder [129] is an approach that automatically reveals relationships between two known objects and displays them as a graph. Since the graph that visualizes the relationships can still become large, interactive features and filtering options were added to the user interface that enable a reduction of displayed nodes and facilitate understanding.

Explorator [65] is an open-source exploratory search tool for RDF graphs, implemented in a direct manipulation interface metaphor. It implements a custom model of operations and also provides a Query-by-example interface. Additionally, it provides faceted navigation over any set obtained during the operations in the model that are exposed in the interface. It can be used to explore both a SPARQL endpoint as well as an RDF graph in the same way as "traditional" RDF browsers.

3.1.2 Query writers

Some system focus exclusively towards the support of novice users for the compilation of SPARQL queries aiming to help them to fully explore the available information. Such system, fail to provide the right functionalities for exploring, understanding and visualizing the data. Novice users can find the complexity of the SPARQL language overwhelming and disengaging. Tools have been developed that help the users form queries while learning the query language.

Konduit VQB [10] provides a way for users to build SPARQL queries in an intuitive way, with having no or little knowledge about the querying language. This does not mean a complete abstraction from the underlying details, but provides an interface that suits the needs of both novice and expert users. SPARQL Builder [336] is an intelligent tool by which users with no knowledge of SPARQL can generate SPARQL queries and retrieve results satisfying their requirements. SPARQL Builder collaborates with TogoTable, a web application enabling biological researchers to upload their data in a table form and add annotations obtained from SPARQL endpoints.

MashQL [217] is a query-by-diagram language that regards the Internet as a database and generalizes the idea of mashups. People are allowed to build data mashups diagrammatically. MashQL queries are translated into and executed as SPARQL queries. The novelty of MashQL is that it allows querying a data source without any prior understanding of the schema or the structure of this source.

SparqlFilterFlow [120] is an approach for visual SPARQL querying based on the concept of extended filter and flow graphs. In contrast to popular approaches, the queries can be created entirely with graphical elements.

SMART [20], Semantic web information Management with automated Reasoning Tool, aims to provide intuitive tools for life scientists to represent, integrate, manage and query heterogeneous and distributed biological knowledge. Features include semantic query composition and validation, a graphical representation of the query, and the retrieval of pre-computed inferences from an RDF triple store.

3.1.3 Schema extraction

Other systems focus on the schema identification. Such techniques often fail due to the lack of consistency of the underlying data, or extract unreliable schemas based only on a small subset of the dataset. In addition, such systems do not offer any functionalities for accessing the data. Recognising that significant effort has been dedicated to the visualization of data from relational schemas, many tools try to extract the SPARQL endpoint schema aiming to re-utilize the available techniques.

TBox visualization [327] aims to extract and visualize the information on the used schema, also called TBox from SPARQL endpoints. Rather than relying on given

TBox information, the tool infers what a TBox for the available ABox data could reasonably look like based on several SPARQL queries. This information is incrementally added to an interactive graph visualization based upon the Visual Notation for OWL Ontologies. A node-link-based graph visualization is chosen, as it allows users to grasp certain structural criteria at a single glance, such as the presence of highly linked central classes or largely disjoint clusters of classes, before proceeding to a deeper analysis.

ViziQuer [343] asks the user to provide an address of a SPARQL endpoint that is of interest, then it extracts and visualizes graphically the data schema of the endpoint. The user is able to overview the data schema and use it to construct a SPARQL query according to the data schema. The tool extracts a simplified data schema by using a predefined sequence of SPARQL queries at the SPARQL endpoint. This process can take a while since schema retrieval depends on ontology size and speed of the SPARQL endpoint while only typed data are supported.

Afterburner [80] implements an analytical RDBMS in pure JavaScript so that it runs completely inside a browser with no external dependencies. It generates compiled query plans that exploit two JavaScript features: typed arrays and asm.js. Afterburner has the ability to support interactive data exploration via automatically-generated materialized views.

3.1.4 Semantic Similarity

There are many techniques developed that aim to calculate the semantic similarity between set of words, that take advantage of the topological similarity of the ontologies or the pair similarity of words.

As already discussed, words are connected with linguistic relationships forming a network of terms that can provide information regarding the semantic similarity. These relationships are used by many systems that perform semantic analysis. In many cases, such as in [243], the systems utilize the topology of the words to demonstrate that semantic similarity between nearest neighbors can be used for the classification of words.

Many systems, exploit the 'is a' semantic relationships to measure semantic similarities, such as in [254], where the notion of information content is exploited to improve the edge counting method and overcome the problem of varying link distances.

In other cases, additional information is used to augment the topological information. As an example, in [196], an information-theoretic measure of semantic similarity that exploits both the hierarchical and non-hierarchical structure of an ontology is presented. This measure addresses the general question of how text and link analyses can be combined to derive measures of relevance that are in good agreement with semantic similarity.

In addition to the topological similarity, many systems, such as [244], exploit a sense-based probabilistic representation that enables detection of the similarity between the meanings of the words. This enables them to compare different types of linguistic data and operate at multiple levels, from comparing word senses to comparing text documents.

Further focusing on the context of the information is the context-aware solution for the semantic similarity measure in the ontology environment, presented in [74].

This solution contains an ontology conversion process and a hybrid semantic similarity model, which involves assessing the concept similarity from the perspectives of both the ontology structure and the context of ontology concepts and relations.

A different approach for measuring semantic similarity between words and concepts is presented in [152]. This approach combines a lexical taxonomy structure with corpus statistical information so that the semantic distance between nodes in the semantic space constructed by the taxonomy can be quantified with the distributional analysis of corpus data. This approach is further developed in [208] where the semantic similarity of texts is measured using in addition to corpus-based and knowledge-based measures of similarity.

Here, special focus is given to the adaptation of the metrics for short texts which are nowadays widely available due to the social media interactions. A knowledge-based method for measuring the semantic similarity of texts is presented in [57]. An algorithm that combines the word-to-word similarity metrics into a text-to-text semantic similarity metric is presented, showing that this method outperforms the simpler lexical matching similarity approach.

Regarding the calculation of the semantic relatedness between sets of words, there is limited research. As already discussed, this is due to the fact that in many cases the term semantic relatedness is used interchangeable with the semantic similarity and not taken into consideration. Explicit Semantic Analysis [104], is one of these research studies, that proposes a novel method that represents the meaning of texts in a high-dimensional space of concepts derived from Wikipedia. Machine learning techniques are used to explicitly represent the meaning of any text as a weighted vector of Wikipedia-based concepts. Assessing the relatedness of texts in this space amounts to comparing the corresponding vectors using conventional metrics.

A different measure of semantic relatedness is proposed in [22], which examines two concepts and extracts a measurement based on the number of shared words in their definitions. This measure extends the definitions of the concepts under consideration to include the definitions of other concepts to which they are related.

Another approach [334], uses Wikipedia to provide structured world knowledge about the terms of interest. Aiming to improve similar solutions, this approach does not examine category hierarchy or textual content, but focuses instead on the hyperlink structure of Wikipedia.

Finally, a novel method for measuring semantic relatedness using semantic profiles constructed from salient encyclopedic features is presented [125]. The model is built on the notion that the meaning of a word can be characterized by the salient concepts found in its immediate context. SHCNN [153] aims to leverage representation learning for conversation disentanglement. A Siamese hierarchical convolutional neural network, which integrates local and more global representations of a message, is first presented to estimate the conversation-level similarity between closely posted messages. With the estimated similarity scores, the algorithm then derives conversations based on high confidence message pairs and pairwise redundancy.

DeepQA [94] is created by IBM and is widely known by its implementation, Watson a system that is performing at human expert-levels in terms of precision, confidence and speed at the Jeopardy! Quiz show. It is based on a deep analysis of the natural language over a complex network of knowledge database and it handles the problem as a massively parallel hypothesis generation and evaluation task.

3.1.5 Summarization

The systems that use summarization methods [48] can be divided in three basic categories: pattern mining, statistical and structural. Pattern mining methods employ aggregations and graph structures to identify trends in the datasets. Due to the strictness of these trends, such methods are ideal for schema identification. Statistical methods provide quantitative results over the data based on targeted queries and available semantic information. Such methods are used for the selection of the proper dataset for the user needs.

The structural methods create the summaries based on the graph structure and can be further divided in quotient, which aim to identify equivalent nodes based on an equivalence relation over them, and non-quotient that use other structural measures, such as centrality, to create the summaries. Quotient summaries target indexing and querying, while non-quotient summaries are better suited for visualization and data understanding [48]. Thus, we focus further on them.

The Grouping Nodes on Attributes and Pairwise Relationships (SNAP) [302] method is the most well known among them. It focuses on the construction of a graph visualization that uses super-nodes, nodes that contain multiple nodes of the input graph, to create summarizations based on user input and structural information such as edge values and node connections. The main drawback of this solution is the requirement for the user to select the summarization properties in order to produce the visualized graph. Such a limitation is hindering for inexperienced users or users that want to explore datasets they are unfamiliar with.

3.1.6 Community Detection

An alternative to summarization, and a promising solution for intuitive exploration of RDF datasets, is community detection. As discussed in [123], community detection has a key role in the analysis of complex networks and the inference of useful insights regarding graph topology. However, although traditional community detection methods are very useful when applied to small networks, they cannot scale for networks of modern size as they rely on heavy computations and require a significant size of main memory. Therefore, they can process networks of up to only a few thousand nodes and edges. Hence, in order to apply community detection to RDF datasets, we need new scalable and efficient algorithms that use persistent memory and data management models to process larger graphs.

Although community detection lacks a formal definition, it is a well studied concept in the complex network analysis field. Groups of actors (i.e., people in online social networks, smart object, etc.) that interact together tend to form communities, which are groups of actors that interact more closely among them than with the rest of the actors. Community detection (or classification) is an unsupervised learning technique and several algorithms have been proposed in order to find the best set of communities according to several criteria [123]. Hierarchical community detection is a popular class of algorithms.

These algorithms begin by considering a single community consisting of all the nodes and proceed to split the graph into smaller groups until a condition is satisfied [100]. Among these algorithms, Girvan-Newman [112] is one of the most well-known ones. This algorithm, computes the edge betweeness centrality metric [40] for each edge of the corresponding graph and remove the edge with the highest score. This *process is repeated until the graph is split into the required number of communities. Although this algorithm manages to detect communities resulting in high modularity [225] scores, the computation of the edge betweenness centrality score for every edge is rather costly and deems the approach unpractical for modern large graphs corresponding to complex networks of big data scale size in terms of nodes and edges among them.*

In order to deal with this issue several approaches have been proposed, including both centralized and decentralized ones, many of which employ techniques like MapReduce in order to cope with the with the large-scale data [127, 169].

3.2 Data Visualization

3.2.1 Graph-based visualization systems

A large number of systems visualize linked datasets adopting a graph-based approach [202]. In order to cope with large datasets some approaches [19, 73, 294] use sampling and aggregation techniques to visualize what they interpreter as important information while others [89, 339] visualize a limited number of elements based on the user exploration. Such systems are often available as over the web and provide many filtering functions that can be customized.

They present, however, a limited part of the information to the user hindering the overall understanding of the available information and causing some information loss. Additionally, their summarization techniques often restrict the exploration of the data.

For some datasets the relationships between the entities as well as the hierarchy and overall data structure are very important. Such examples are biological data and social media interaction datasets as their analysis is based on understanding the connections between entities.

SynopsViz [31] is a web-based visualization tool that takes into consideration the available hierarchies thus allowing efficient personalized multilevel exploration over classes and properties. In order to provide scalability under different exploration scenarios, the model offers a method that incrementally constructs the hierarchy based on user's interaction, as well as a method that enables dynamic and efficient adaptation of the hierarchy to the user's preferences.

LODWheel [291] investigates new ways of visualizing linked data in graphs and charts so as to be informative to users with little or no knowledge within the domain, as well as to more experienced users.

Lodlive [45] are exploratory tools that allow users to browse linked data using interactive graph navigation. Starting from a given URI, the user can explore linked data by following the links.

ZoomRDF [339] employs a space-optimized visualization algorithm in order to display additional resources with respect to the user navigation choices.

FlexViz [89] offers node and edge specific filters that are based on search and navigation criteria aiming to reduce the amount of handled data and provide to a meaningful subset to the user.

KC-Viz [215] exploits an innovative ontology summarization method, where it becomes possible to navigate ontologies starting from the most information-rich nodes.

LaGO [341] allows straight-line graph drawings to be rendered interactively with adjustable level of detail by combining edge cumulation with density-based node aggregation and exploiting common graphics hardware for speed.

3.2.2 Hierarchical visualization systems

Another approach is the visualization of the information based on hierarchical models. The most well-known and popular techniques, such as the ones presented in [3, 13, 17, 156], present the input dataset either filtered or aggregated, in order to limit the volume of nodes to be handled.

Although the hierarchical approaches provide interactive visualizations with low memory requirements, they have two main drawbacks. Their applicability is heavily based on the particular characteristics of the input dataset as they can be applied only on acyclic datasets that conform to a hierarchical data model. Moreover, most of the hierarchical approaches result in loss of information either because they use aggregation techniques to limit the input dataset size or because they allow navigation only along the hierarchy making information inaccessible from nodes at the same hierarchy level.

While the navigation of highly aggregated data might help initially the user to better understand the structure of the information, at a deeper exploration of the dataset the loss of information might be proven hindering to the usefulness of the visualization. In many cases, such as the analysis of financial datasets, users are more interested in seeing an overview of the information based on semantic categorization and hierarchy. As an example, a financial advisor may be interested to know that stock market prices from telecommunication companies are higher than for banks.

ASK- GraphView [3] is one of the few hierarchical systems that does not require the input dataset to have any specific feature. The clustering, though, is done randomly using a partitioning algorithm that is based exclusively on the morphology of the graph and not on the semantics. This raises the question of the grouping quality and how easy it is for a user to locate specific information. This tool can handle graphs with 16M triplets but only 4K of them are presented to the user after the clustering.

GrouseFlocks [13] works only on datasets structured over well-defined hierarchies. It requires complete data and can handle up to 220K elements when the hierarchy is predefined.

Tulip [17] develops a technique to solve the problem of presenting to the users the data in only one predetermined way, regardless of the features of the input dataset. This tool stores the input data once and extracts the information with different techniques providing the user with multiple views/clusters over the data.

GMine [156] uses the state-of-the-art partitioning algorithm METIS [161] to split the input dataset in reasonable partitions. Each partition is visualized as one node, resulting in the loss of the information included in each partition.

3.2.3 SPARQL endpoint visualization tools

Given that most RDF datasets are available through dedicated SPARQL endpoints, some systems, such as [114, 35] aim to take advantage of the querying capabilities *of the endpoint, limit the volume of the data that they handle and present to the user targeted visualizations for the results. Such systems, can scale for any dataset, be accessible over the web and provide data exploration and querying support. Their main drawback is that users with limited knowledge about the underlying schema may be unable to retrieve information they are interested in.*

Large datasets may contain information about different topics. Exploring and understanding the complete information is not always meaningful. As an example, most users interested in exploring Wikidata, are looking for answers to specific queries. To cover this need, tools have been developed that visualize SPARQL query results.

Visualbox [114] is a system that makes it easier for non-programmers to create web visualizations based on Linked Data. Visualbox provides a unified environment that supports the whole process of creating a visualization based on a SPARQL query. It runs a query on the server and provides a useful caching mechanism that allow users to visualize the data even if an endpoint is down or unresponsive.

The Linked Data Query Wizard [139] is a web-based tool for displaying, accessing, filtering, exploring, and navigating Linked Data stored in SPARQL endpoints. The main innovation of the interface is that it turns the graph structure of Linked Data into a tabular interface and provides easy-to-use interaction possibilities by using metaphors and techniques from current search engines and spreadsheet applications that regular web users are already familiar with.

SPARQL-visualizer [35] aims to facilitate the design process of a shared ontology, where domain experts, software developers and ontology engineers collaborate. As they typically have a different view on the ontology and understanding of the technology, it can be difficult to communicate proposals within the group. Sample data, queries and results of them are visualized in table or graph form to support their collaboration.

3.2.4 Facet browsers

Some systems extract the semantic information from datasets aiming to support their exploration with the use of facets. Facets are a subset of filtering, that help the users quickly identify their filtering options without navigating the complete information. Such filtering is very useful when the user wants to search for something very specific but fail for general criteria or soft categorizations.

In many cases, users are interested in combining the available information based on its semantic annotations or other filtering approaches. An indicative example users may be interested in temperature readings, but they want to specific or convert the retrieved information to a specific metric system avoiding any confusion between Celsius and Fahrenheit degrees.

gFacet [131] is a browsing approach that supports the exploration of the Web of data by combining graph-based visualization with faceted filtering functionalities. The graph-based visualization facilitates a comprehensible integration of different domains; the use of facets supports a controlled filtering of information. With gFacet, users are enabled to browse the Web of data efficiently and to retrieve information from different user-defined perspectives.

Facet Graphs [128] allows humans to access information contained in the Semantic Web according to its semantics and leverage the specific characteristic of this

Web. To avoid the ambiguity of natural language queries, users only select already defined attributes organized in facets to build their search queries. The facets are represented as nodes in a graph visualization and can be interactively added and removed by the users in order to produce individual search interfaces. This provides the possibility to generate interfaces in arbitrary complexities and access arbitrary domains.

3.3 Spatial Data Management

3.3.1 Spatial Indexing

R-tree. *The R-tree spatial index was first proposed by Antonin Guttman in 1984[117]. It is a height-balanced tree where the data objects are indexed with pointers from the leaf nodes, designed so that spatial searches require visiting only a small number of nodes and leaves. The index is very similar to the B-tree index [53, 21], meaning that it is a dynamic data structure where insertions and deletions can be handled along with searches without requiring any periodic reorganization.*

The key concept for the R-tree is that objects that are in close spatial proximity can be grouped together and represented with their minimum bounding rectangle in the next higher level of the tree. A very important parameter for the efficiency of the R-tree index is the quality of the minimum bounding rectangles, measured by examining if they are covering empty space or they are overlapping.

As a result, there is no guarantee regarding the worst-case performance during a search transverse. The search algorithm begins from the root as with the B-tree but in some cases more than one sub-tree under a node must be visited based on the search criteria and the data indexed.

There are two important disadvantages concerning the R-tree indexes. To begin with, when the search query is for a location point, it is highly likely that this point will be included in multiple minimum bounding rectangles, leading to the investigation of several paths, and minimum bounding rectangles, from the root to the leaf level. This issue become more prominent when there is significant overlap between the rectangles. The second disadvantage is the fact that only a few large rectangles are enough to significantly increase the degree of overlap, causing performance degradation during range query execution, due to empty space. Based on the indexed data, objects that are significantly larger than average size of the dataset may easily case such issues.

Formally defined, the performance of a R-tree index is dependent on the minimal coverage and overlap. The term coverage refers to the area that all the nodes of the tree cover, while overlap is the area that belong to more than one tree node. Minimal coverage refers to the effort to reduce the not-used space, the empty space that it is included in the nodes of the tree but not in any of the indexed spatial data objects. Minimal overlap is the effort to minimize the overlap between nodes of the same tree level, reducing the number of possible sub-trees matching a query and the search paths to the leaves. In order to achieve the highest efficiency for the search requires both minimal coverage and minimal overlap are needed.

Despite these challenges, and based on the fact that it has been proved that R-trees scale efficiently for real-world data [144], they have found a plethora of uses in many domains. As an example, R-trees are used to improve the nearest neighbor *search algorithms for many distance metrics, a feature used by many IoT sensors when they want to connect with their nearest sensors in order to become aware of the current status of their environment. Another noticeable use of the R-trees is to store sensor locations as spatial objects ranging from points to complex polygons, depending on the sensor types and the measured parameters, annotated into multiple data categories based on their characteristics allowing the visualization of the IoT systems in the two-dimensional space. R-trees are also very popular for navigation systems as they can answer efficiently spatial queries that are of interest to most users, such as the nearest pharmacy or all the super markets within a given radius.*

R-tree variations. *Given the popularity of the R-tree indexes, their performance in real-world data and their efficiency there were many research efforts focusing in ways to improve their performance, aiming mostly to resolve the two main disadvantages discussed above. Some of the proposed solutions managed to offer a worst-case performance limit and improve the efficiency of the index for point location queries. They were however significantly more complex and difficult to use in everyday application so they never gained in popularity. The most well-known of these variations are the following:*

R$^+$ tree. *R$^+$ tree index is trying to resolve the minimal overlap issue of the R-tree index, the overlapping between minimum bounding rectangles in the internal nodes of the tree, by allowing an object to be included in more than one leaf of the tree as needed and by not allowing overlaps between nodes of the same tree level [272]. This decision has as a result to reduce the overlap area of the nodes, reducing the number of sub-trees accessed for each query, increasing the efficiency of the index and minimizing the response time.*

The main differences between the R-tree and the R$^+$ tree can be summarized as the lack of overlaps in internal nodes of the same tree level, the indexing of a data object more than one in the leaves and the fact that nodes are no longer required to be half filled. This differentiation, increases the point location query performance, as the spatial region of interest is included at most in one node of each tree level.

R$^+$ tree indexes have two main drawbacks. On the one hand, the fact that some data objects are indexed more than once means that the R$^+$ tree is significantly larger than the R-tree constructed over the same data. This may turn into an important bottleneck as the volume of data increases. On the other hand, there is a higher cost to construct and maintain a R$^+$ tree index, making it unsuitable for some application. For example, navigation applications that run on mobile devices that have limited computational power may have performance issues with the use of a R$^+$ tree index.

R$^+$ tree indexes are mainly used for IoT systems that include complex geometries with important differentiations in size and shapes. A characteristic example is industrial production networks that include objects ranging from point sensors, such as thermometers, to area sensors, such as thermal cameras that should be combined and integrated.

R* tree. *R* tree index aims to achieve both the minimal coverage and the minimal overlap by implementing a revised node split algorithm and introducing the concept of forced reinsertion at node overflow[24]. This approach is based on the fact that the quality of a R-tree index is highly depended not only on the spatial objects that* *are indexed but also on the order in which these objects are inserted to the index. Rather than creating an index using a bulk-load technique, the R^* tree index performs a series of deletions and re-insertions of data objects aiming to relocate indexed object to more appropriate sub-trees and leaves, thus increasing the performance of the index.*

The node re-insertion has been designed as an optimization process that it is triggered on node overflow. In detail, each time a node overflows rather than causing a node split, some of the data objects indexed by this node are deleted and re-introduced to the index. In order to avoid an infinite deletions and re-introductions loop, in case that there is no improved indexing, the optimization is run only once per tree level upon the insertion of a new data object.

The R^* tree index provides an optimized R-tree without affecting the worst-case query and deletion complexity while at the same time increasing the insertion complexity. This is the reason why the R^* tree indexes are not commonly used in real-world applications.

Hilbert R-tree. *Hilbert R-tree [159] is yet another variation of the R-tree, this time incorporating elements of the B^+ tree index [53] in the original R-tree index, aiming to index multidimensional spatial data objects.*

Given that the quality of the R-tree is dependant on the minimum bounding rectangles created at the tree nodes, the Hilbert R-tree index uses the space-filling Hilbert curve [135] to impose a linear ordering on the spatial rectangles.

Hilbert R-trees have two versions, one for static data when updates, insertions and deletions are minimal or not happening at all and one dynamic, that allows real-time data modification. In both versions Hilbert space-filling curves are used aiming to offer an improved ordering of the spatial data objects in the tree node. The ordering aims create tree nodes that contain such spatial objects that adhere both the minimal coverage and the minimal overlap needs.

The dynamic version of the Hilbert R-tree index introduces a flexible deferred splitting mechanism aiming to increase space utilization. The key idea is based on the sibling nodes that are defined for each node of the tree. The identification of the sibling nodes is based on an ordering of all the nodes of the R-tree. In detail, each rectangle is represented by the Hilbert value of its center, the length of the Hilbert curve from the center of the root node to the center point of the rectangle, which is then used to order all the rectangles.

Based on this ordering, every node has a strictly defined set of sibling nodes that are used for deferred splitting. This allows the Hilbert R-tree index to reach any degree of space utilization, in complete contrast with the traditional R-tree indexes that have no such control.

The static version of the Hilbert R-tree is the most commonly used at IoT applications where the spatial data are complex polygons, as is the case with camera coverage maps, two-dimensional representations of the areas covered by a monitoring camera system.

X-tree. *The eXtended node tree, X-tree, index [27] is designed to improve the performance of the R-tree regarding the query processing for high-dimensional spatial data. It supports datasets that contain both point and extended spatial data. The X-tree was designed based on the observation that for low-dimensional spatial data the* *most efficient organization of the index is a hierarchical approach such as a balanced R-tree while for high dimensional spatial data a linear organization of the index is more efficient, given that due to the expected high overlap, most of the index will be searched for each search query.*

The X-tree index aims to offer minimal overlap by balancing the linear with the hierarchical organization of the index. In detail, the X-tree index employs a two step process for managing overflown nodes dynamically that ensures that the data objects which produce high overlap are organized linearly while those that do not have too much overlap are organised hierarchically. Initially the index employs a split function that divides the overflown node if and only if a split devoted from overlap is possible without allowing the tree to degenerate.

In detail, in many cases due to the restriction of the lack of any overlap, the split may be unbalanced with one of the nodes to be almost full while the other almost empty. In such cases, that decrease the storage utilization the node is not split. In the cases that a split is not possible or efficient, then the X-tree index allows the nodes to include more objects than their initial size, creating what is called super-nodes. Super-nodes are defined as extended directory nodes of variable size.

As a result, based on a balance between not allowing any overlap when splitting overflown nodes and the use of super-nodes with an increased capacity, the X-tree index automatically offers a hybrid organisation of the index that it is as hierarchically as possible without compromising the efficiency of the index. As the IoT increases in popularity and more datasets become available, the need for an efficient indexing of high-dimensional data has become increasingly prominent. The X-tree vector is used in excess in many applications, including map datasets, 3D object designs, string matching and document search, where high-dimensional feature vectors are used with significant query needs.

Quadtree. *A quadtree [96] is a tree data structure in which each internal node has exactly four children. Quadtrees are focusing on the two-dimensional space, while octrees, which we will discuss in Section 3.3.1 are focusing of the three-dimensional space. The idea behind the quadtree index is to recursively divide a two-dimensional space into four quadrants or parts.*

There are no specific requirements regarding the properties of the quadrants, which can be any two-dimensional spatial object including random polygons, square or rectangular, as long as they lead to an indexing of the spatial data in a way that spatial objects included in a leaf node are spatially related based on the query needs of the dataset.

In detail, it is very straightforward to create a quadtree spatial index. The index starts with the root node that is covering the complete area within all the data objects to be indexed belong. Below the root there is the first level of internal nodes, that must have exactly four children, one for each quadrant obtained by dividing the area covered in half along both axes. The tree may have multiple levels of internal nodes all following this rule. Finally, there are the leaf nodes that contain one or more indexed spatial objects.

The insertion of data into a quadtree follows a process similar with the R-tree hierarchical data structure. The data object is examined against the root and follows the path through the intermediate nodes till the correct leaf node. The spatial object is added to the node if there is enough space or triggers a split process.

There are two key parameters that affect the performance of the quadtree index, the dimensions of the quadrants and the size of the leaf node. The two parameters are dependent on the use case, the size of the dataset and the update rate of the information while they affect in addition to the performance of the index when retrieving information, the size of the index and the memory usage, the complexity of the implementation and the overhead for inserts, updates and deletions.

Octree. *As implied by its name, an octree [203] is a tree data structure where each internal node has exactly eight children. Similarly with the quadtree index that we discussed above, the octrees are used to index spatial objects at a three-dimensional space by recursively subdividing the area of interest into eight octants or parts.*

The process of creating a octree spatial index is more complex than the one for the quadtree as here the space is not split based on the axes. In the octree each intermediate node stores an explicit three-dimensional point, defining the point region of the node, that represents the center of the area indexed by the node, and its subtree. It is designed so that the root of the tree represents the infinite space.

It is worth noting that the octree index differentiates from both the quadtree, that we discussed in Section 3.3.1, and the kd-tree, that we will discuss in Section 3.3.1, as the octree indexes split the overflown nodes based on points of the three-dimensional space rather than the dimensions.

Binary space partitioning (BSP) Tree. *Binary space partitioning (BSP) [270, 101] is referring to the process of dividing a space into two parts with the help of a hyperplane, a subspace with one dimension less than that of initial space that it is indexed. This method of space partitioning is used to create a tree hierarchical data structure, where each intermediate node of the tree represents the hyperplane that divides the space into two convex sets allowing the indexing of spatial data objects.*

Binary space partitioning has been mainly used for 3D computer graphics and especially when there is a first-person player perspective, the gamer is playing the game from the main character's viewpoint. In such gaming scenarios, there is a need to efficiently create three-dimensional scenes that include many spatial objects and game elements in a varying degree of distance from the player.

The most popular technique for these scenes is the painter's algorithm [222], which produces polygons in order of distance from the viewer, back to front, painting over the background and previous polygons with each closer object. Using this method has to important issues, in most cases it is very inefficient and time consuming to order the polygons based on their distance from a given point and there may be issues with the overlapping objects.

BSP indexing supports both the efficient retrieval of only the spatial objects, game elements, that are within the frame of the gamer and also their ordering, with linear cost to the number of polygons in the scene, with regards to the distance of the objects to the gamer's position in the scene. In addition, overlapping polygons are subdivided to eliminate possible errors allowing the developers to efficiently use the painter's algorithm to create the scenes. The main drawback of this index is that its contraction is very time-consuming, as a result using the index requires some pre-processing steps, a large portion of the indexed items should be static and the number of moving elements indexed should be minimal.

Binary space partitioning is a generalization of the k-dimensional trees, that we discuss in Section 3.3.1, and the quadtrees, that we discuss in Section 3.3.1. There main differences can be summarized in that they do not enforce any restriction to the number of children for the intermediate nodes and that there are not restriction regarding the hyperplanes, the dimension used or their orientation. In many cases, the hyperplane used by an intermediate node may even be the same as on of the spatial data objects that are indexed.

k-dimensional tree. *The k-dimensional tree index [25, 26, 213] is a differentiation of the binary space partitioning trees, which we discussed in 3.3.1, focusing on organising and indexing point locations in the k-dimensional space. The k-d index has a very targeted application space, specifically in range and k-nearest neighbor searches when the search parameters are k-dimensional spatial objects.*

The k-d tree is based on the creation of hyperplanes, a subspace whose dimension is one less than that of its ambient space, in order to achieve space partitions. Each of the intermediate nodes of the tree create a hyperplane that divides the space into two parts, known as half-spaces. Similarly to other tree data structures, the spatial location points that are indexed and belong to the left of this hyperplane are located in the left sub-tree of that intermediate node while location points that belong to the right of the hyperplane are indexed under the right sub-tree.

In the k-dimensional space a hyperplane that splits the space in two parts can be created in any of the k-dimensions with a hyperplane that it is perpendicular to that dimension's axis. During the creation of a k-dimensional tree, the intermediate nodes are not using the same dimension for splitting the multidimensional space but each one is associated with one of the k-dimensions.

Grid spatial index. *The grid spatial index creates over the two-dimensional space a grid similar to the reference grid that can be found on a common road map. The grid is used to index the spatial data objects into the right grid positions. Depending on the size of the area to be indexed, the number of spatial objects indexed by each node and the available memory the index may have more than one grid level, resulting in a hierarchical data structure, where the area covered by each grid cell becomes smaller towards the grid level that stores the indexed objects.*

The major difference of this index with regard to all the others that we have discussed so far is that the grid spatial index is a data independent method, where the number of grid cells as well as the size of the area they cover, the levels of the index and the overall area cover by the grid are pre-defined and do not change dynamically based on the data that are to be indexed.

This difference leads also to the most critical advantage of the grid spatial index, as the structure of the index is created independently of the data, the way that the data introduction is performed does not affect the efficiency of the index, an issue that we discussed in relation with the R-tree and that the R^* tree tries to eliminate with forced data object re-introduction. It also allows the parallelization of tasks, with multiple instances of the index with the same structure receiving data at the same time, as their merge is very easy and straightforward.

For these reasons the grid spatial index is very popular in IoT systems that are distributed over spatial zones, such as storage facilities and distribution centers as it is very efficient for accessing all the collected information for the area of interest.

Geohash. *Based on the work presented by G. M. Morton [214], Geohash [108] is a public domain geocode system which encodes any geographic location, by encoding each pair of latitude-longitude into a short string of letters and digits. The main idea behind Geohash is to mimic the space-filling curves by creating a grid network over the space that it is to be index and a node for each place of the grid. The nodes are filled with spatial objects that belong in that part of the space. The index forms a hierarchical spatial data structure with each part of the encoded string better specifying the place on the grid of the spatial data object.*

Geohash index have some interesting properties that include arbitrary precision, the ability to remove characters from the end of the encoded string to reduce the size of the index with the trade off of losing the precision. Geohash indexing also guarantees that the length of the shared prefix between two encoded strings is an indicator of their spatial proximity.

On the other hand, two completely different string do not guarantee that the spatial objects are spatially distant. Two location points can be in adjacent grid buckets and still have no shared prefixes between their encoded strings. Geohash indexes are used mainly in IoT systems that are distributed over large geographical areas, an indicative example are systems that are deployed in natural ecosystems to record environmental parameters.

HHCode. *An early version of the HHCode index was initially designed and developed by scientists working for the Canadian Hydrographic Service's Atlantic regional offices at the Bedford Institute of Oceanography in Dartmouth, Nova Scotia [318] as a solution to the problem of storing the huge spatial datasets collected while conducting hydrographic surveys. These surveys had a very strong temporal element in addition to the spatial one that triggered the idea of a spatio-temporal indexing system based on the Riemannian hypercube data structure, a helical spiral through the three-dimensional space that allows n-size feature vectors.*

In practice, the HHCode index [164] uses binary helical hyperspatial code to represent multi-dimensional spatial data. In the binary HHcode data structure the data objects are represented using the BH code over a N-tree structure derived using recursive decomposition. The data structure has been designed to maintains the dimensional organization of multi-dimensional data. The index defines an upper limit of data objects that can be index by each index node. If a node exceeds this limit then it is partitioned into two children nodes while the parent node is not retained. A dedicated data structure stores the information about partitioned node and related BH code values.

3.3.2 Spatial Data Management Systems

Relational Databases. *A relational database is a digital database based on the relational model of data, as proposed by E. F. Codd in 1970 [52]. The relational database systems are using the Structured Query Language (SQL) [37] for querying and maintaining the database.*

Initially, a relational databases was defined as a database system that was complying with Codd's 12 rules [184] but these rules were proven too complicated for commercial implementations [61], so practically a database management system is relational when the data are presented as relations and can be processed using re-

lational operators. Most relational database systems were not initially supporting spatial data, but they were updated to do so as the needs arose. Indicative examples of such implementations are presented here.

MySQL [330] is a relational database management system written in C and C++ provided as a free and open-source software under the terms of the GNU General Public License. MySQL has based the design of the spatial support on the Open Geospatial Consortium [231] standard proposal, the OpenGIS Implementation Standard for Geographic information - Simple feature access - Part 2: SQL option[134], that proposes several conceptual ways for extending an SQL RDBMS to support spatial data [277]. MySQL offers support for point, lines and polygons spatial data types as well as for the more generic geometry spatial data type.

MySQL can create spatial indexes [218] using syntax similar to that for creating regular indexes, with the addition of the SPATIAL keyword for differentiation. An important requirement is that columns containing spatial objects that are to be indexed must be declared not null. In MySQL the created spatial index is either an R-tree index, or B-tree index but only for storage engines that support non-spatial indexing of spatial columns.

PostgreSQL [290] is a free and open-source relational database management system that was designed to emphasize extensibility and SQL compliance. The system was extended in order to provide spatial support, creating the PostGIS [230] spatial database extender. PostGIS does not use the R-tree to index the spatial data objects [247] but the Generalized Search Tree (GiST) index [132]. The GiST index is not dedicated to the indexing of spatial information, but an extensible, disk-based index structure for large data sets, that offers a general-purpose implementation with a simple design.

H2GIS [119] is the spatial extension of the H2 database engine [118]. The H2 is a relational database management system written in Java designed to be embedded in Java applications or run in client-server mode. H2GIS offers support for point, lines and polygons spatial data types and implements the functions specified by the OpenGIS Simple Features Implementation Specification for SQL. The current release of the implementation does not offer full support of the spatial indexing and no details about the indexes used are provided [146].

IBM Informix [285] is a fast and scalable database server that manages traditional relational, object-relational, and web-based databases. Informix supports alphanumeric and rich data, such as graphics, multimedia, geospatial, HTML, and user-defined types. The server provided spatial support for relational tables, allowing columns with spatial data types and offering R-tree indexing for the spatial information.

There are two interesting facts about the behavior of the R-index at IBM Informix server [286], the first one being that a database cannot be renamed if it contains an R-tree index, because R-tree indexes are implemented with secondary access method. The second has to do with the query optimizer that will not use the R-tree index unless the statistics on the table are up-to-date.

Last but not least, Microsoft SQL Server [274] is a relational database management system developed by Microsoft. It supports the most spatial data types [275] from any other implementation discussed so far, including spatial shapes such as circular string, compound curve and curved polygon.

The Microsoft SQL Server does not implement any of the spatial indexes that we *discussed above. Here, the spatial indexes are built using B-trees, facing the challenge of representing the 2-dimensional spatial data in the linear order of B-trees. In order to overcome this issue, the index is contracted before any data are inserted into the data structure. The index is implemented over a hierarchical uniform decomposition of space. Specifically, the space is decomposed into a four-level grid hierarchy, where each successive level further decomposes the level above it [276].*

Key-value Databases. *A key-value database is a non-relational database that uses a simple key-value method to store data. The system handles the data as a collection of key-value pairs where the key serves as a unique identifier. Most implementations do not impose any restrictions to the data types of the keys or the values, which can be anything from simple variable to compound objects. The main advantage to the key-value databases is the fact that they are easily distributed and allow horizontal scaling at scales that other types of databases cannot achieve.*

Remote Dictionary Server (Redis) [47, 265] is an in-memory data structure project implementing a distributed, key–value database. Redis is an in-memory but persistent on disk database, so it offers a balance between very high write and read speed and the limitation that the data sets should fit in the available memory. Redis is different from other key-value solutions as here values can contain more complex data types, with atomic operations defined on those data types.

In addition, Redis data types are closely related to fundamental data structures and are exposed without additional abstraction layers. Redis has several functions related to spatial data objects. The platform encodes the latitude and longitude into the score of a sorted set using the Geohash algorithm.

GeoMesa [110], while technically not an autonomous database management system, is included here as it is an open source suite of tools that enables large-scale geospatial querying and analytics on distributed computing systems [142]. Specifically, GeoMesa provides spatio-temporal indexing on top of the Accumulo, HBase, Google Bigtable and Cassandra databases for massive storage of point, line, and polygon data.

GeoMesa provides a series of spatial indices in order to satisfy various search predicates [146]. There are indexes for two and three dimensional spaces, for time and space simultaneous indexing, for space and additional variables indexing.

In detail, GeoMesa offers a two-dimensional Z-order curve to index latitude and longitude for point data. This index is created for the spatial data type Point. This is used to efficiently answer queries with a spatial component but no temporal component.

GeoMesa also offers a three-dimensional Z-order curve to index latitude, longitude, and time for point data. This index is create when both a spatial data type Point and a time attribute are available. In addition, it offers a two-dimensional implementation of XZ-ordering [39] to index latitude and longitude for non-point data and a three-dimensional implementation of XZ-ordering to index latitude, longitude, and time for non-point data.

Graph Databases. *The term graph database [12] is used for systems that relay on graph structures for semantic queries with nodes, edges, and properties to represent and store data. They were designed to address the limitations of existing solutions regarding the relationships between data objects. Both the relational model and most* *of the NoSQL database models connect the data by implicit connections relationships are implied and must be realized at run-time, such as the external key. The graph model explicitly lays out the dependencies between nodes of data, managing the relationships between data objects as equally important entities that can be labelled, directed, and given properties.*

Neo4j [326, 209] is a graph database management system developed by Neo4j, Inc [220]. Neo4j is the most used and most popular graph database and one of the few of its king that can ensure an ACID-compliant transactional database while maintaining native graph storage and processing. For spatial indexing [221], Neo4j uses space filling curves over an underlying generalized B+Tree. Points are stored in up to four different trees, one for each of the four coordinate reference systems. This allows for both equality and range queries using exactly the same syntax of Cypher queries and behaviour as for other property types.

Oracle Spatial and Graph [287] is high performance, enterprise-scale, commercial spatial and graph database. It supports multiple application related to big data management, spatial information management, information retrieval and data analytics. It offers a general-purpose property graph database and analytic features that are used for the analysis of social networks, the management of Internet of Things systems, smart algorithms for fraud detection as well as decision support and recommendation systems. The system allows either bulk or transactional loading of the spatial information but forces the creation of a R-tree index for the spatial elements to ensure the efficient access to the information.

Document-based Databases. *Document-based databases are designed for storing, retrieving and managing semi-structured data that cannot be stored in a strict relational environment, also thought as document-oriented information, documents that contain data in some standard format or encoding. Such databases offer not only the expected key-to-document lookup functionalities but also query capabilities, through dedicated APIs or query languages, that allow the retrieval of documents based on their content, for example the presence of a property or specific value for a field.*

Apache CouchDB [58] [11] is an open-source document-oriented NoSQL database, implemented in Erlang. GeoCouch [211] is a spatial extension for Couchbase and Apache CouchDB. It takes advantage of the GeoJSON [44] format to provide support for all geometry types including GemetryCollections. An interesting implementation detail is that the key 'bbox' of the GeoJSON geometry takes priority over the calculation of the bounding box of the geometry. This means that is the GeoJSON contains a bbox property it will be used instead of calculating it without any validation actions. GeoCouch supports multidimensional indexing that it is automatically selected based on the content of the documents.

RethinkDB [303] is a free and open-source, distributed document-oriented database that is handling documents containing JSON objects. RethinkDB supports the WGS84 World Geodetic System's reference ellipsoid and geographic coordinate system. It does not directly support any projected coordinate system or the three-dimensional space. RethinkDB [255] offers spatial indexing for the supported spatial objects, points, lines and polygons.

Chapter 4

SPARQL endpoint exploration

We present here a complete solution that supports both expert and novice users with querying SPARQL endpoints, exploring and visualizing the results in a dynamic way. Aiming to create a system that will overcome the challenges mentioned above we developed a client-server architecture that can offers:

Schema agnostic. Our system is carefully designed to handle any endpoint, without any information about the underlying schema, without any compromise on the user experience.

Decision Support System. We have developed a DSS that makes the decision regarding the visualization type that should be used for a query result. The DSS can provide specific parameters and layouts for queries that are to be visualized as graphs and specific charts for queries that contain less variables and aggregated information.

Knowledge database & Experimental analysis. We have accessed and analysed multiple SPARQL endpoints to collect information regarding possible query results. This information is used by the DSS to define the visualization parameters of the graphs. We have also performed a detailed analysis of the results, evaluating the range of parameters for the endpoints and associating them with the corresponding visualization parameters.

Query-specific visualization rules. We have studied the methodologies for choosing the right charts for the right data and we have utilized them to create proper rules that will match query types with the proper visualization chart. We provide case-specific visualizations for query results, based only on information and features extracted from them.

Integrated platform. In order to showcase the flexibility and the robustness of this approach, we developed an integrated platform that allows the users to query a SPARQL endpoint of their preference, either by writing their own query or by using a supportive form or by simply running a keyword search, and visualize the result based on its type and characteristics. Furthermore, the platform allows the user to filter the visualized information dynamically by exploiting the semantic annotations of the data and further explore the dataset by providing support and hints towards the next exploration steps.

4.1 System Architecture

We present the system architecture in Figure 4.1. The system can be divided into two main modules, the integrated platform and the Decision Support System (DSS). The

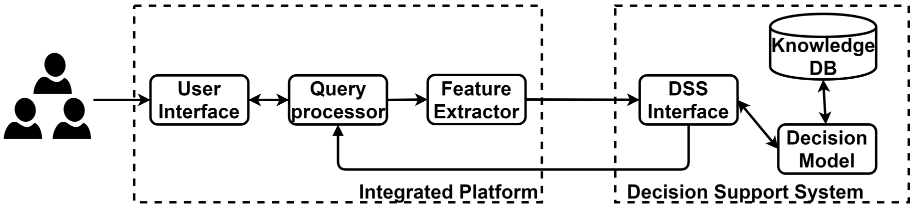


Figure 4.1: System Architecture

integrated platform is the client component, accessible by the user and tasked with all the user-related interactions as well as the communication with the SPARQL endpoint while the DSS is the server component accessible through a dedicated interface. The integrated platform has three sub-modules, the User Interface which is responsible for visualizing the query results and support the user is locating and exploring the information of interest through a series of functionalities, the Query processor which receives the SPARQL query and queries in real time the selected endpoint and the Feature Extractor which extracts the needed features from the result and forwards this information to the DDS interface. The Decision Support System has three key sub-modules, the Knowledge Database that contains the raw data needed to make a decision, the Decision Model that has all the rules and logic of the decision making and the DSS Interface that receives the extracted features from the integrated platform and returns the decision of the DDS in the form of visualization parameters. We present below the modules of the architecture in detail.

4.2 Decision Support System

A DSS is an information system that supports decision-making activities. DSSs are designed to support the operational planning and help people make decisions about problems. Their main contribution is that they can support problems that are rapidly changing and not easily specified in advance. A DSS can be built in any knowledge domain as long as enough information can be collected to support the decision model. A DSS is designed to combine relevant information provided from a knowledge base and models to solve problems and make decisions. There are three key components to a DSS architecture. The knowledge database that should contain data presenting the real word and serve as the basis for the system. The model, the core logic of the system where based on the available information all the decisions are made and the interface where the current problem/situation is given as an input and a decision is returned as an output.

We believe that a DSS can be very useful in the context of the visualization of SPARQL query results. To begin with, the problem of effectively visualizing a specific query result is within the core problems that the DSSs can handle. This is due to the fact that it is a problem that cannot be specified in advance. Two query results are expected to present high diversity regarding their characteristics even for a single endpoint. A DSS provides the flexibility, through the modeling process, to have the needed rules to support such diversity.

Example 4.2.1 *An indicative example for the STW Thesaurus for Economics can be the diversity between a query from a data scientist interested in the number of appearances of each predicates available in the dataset and a query from an economist* *interested in retrieving all the terms containing the keyword Economy and their description. In the first case, the result is showing the distribution of the 100% of the predicates and should be represented as a pie chart while in the second case the result is 121 triples containing terms and their description which should be represented as a graph.*

In addition, modeling data to visualization types using the proper parameters is an intuitive process that follows specific empirical rules when the characteristics and format of the data are known. Last but not least, dynamically offering case-specific visualization parameters for each query result allows the exploration of any SPARQL endpoint without requiring any knowledge for the underlying schema or enforcing any limitation to the exploration.

We have developed a decision model that proposes two visualization categories, graphs for queries results containing triplets of information without a time variable and charts for query results containing one or two variables and aggregation functions. In order to determine the parameters that are needed for the graph layout of a query result we have developed a knowledge database with information from many available endpoints. The knowledge database allows us to determine if a graph is highly connected, if it contains a lot of descriptive information or follows a specific pattern allowing as to choose the right layout algorithm and parameters. Regarding the choice of the appropriate chart we have created a series of rules as part of the decision model that interpreter known data visualization rules to specific query types. We present below the implementation details of the three sub-modules of the DSS.

4.2.1 Knowledge Database

According to studies that were carried out on logs of endpoints, aiming to study patterns in the queries and the users of the endpoints [30, 197, 258], the majority of the queries are SELECT queries for triples without aggregation. This means that the most important part of the visualization that we need to properly parameterize are the graphs. In order to achieve that we need to know what are the expected range for the characteristics of such queries. We have accessed and analyzed multiple SPARQL endpoints to collect information regarding the result size limit, the total number of unique predicates, the total number of unique entities, the most/least connected entities, the most/least used predicate, the min/max string length for entities and predicates and the average node degree. We present in Section 4.4 the details regarding the experimental methodology and results. The collected information allows us to determine if the characteristics of a result are within the expected limits, identify potential issues and choose the right visualization parameters. The information collected is stored in a relational database accessible by the decision model.

4.2.2 Decision model

As presented in Figure 4.2, we have developed a deterministic decision tree that identifies the proper visualization type based on the query type and result characteristics. Triplets of information containing no time information are visualized as graphs. In Section 4.2.2, we present which features of the query result are examined and how they are utilized in defining the visualization parameters for the graphs.

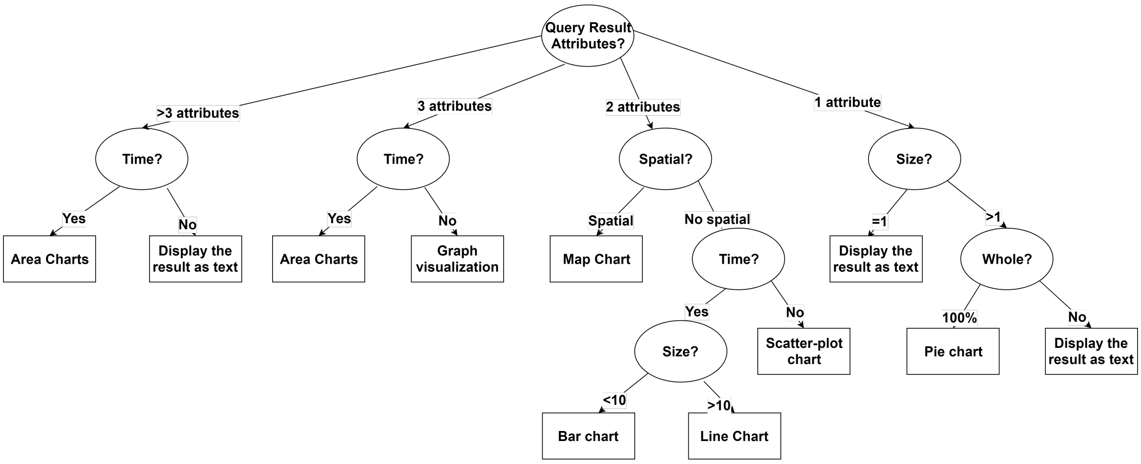


Figure 4.2: The tree showing the decision making process for the model

Here, the information available in the knowledge database is used to support the process and provide the needed information regarding expected values. In Section 4.2.2, we present the decision making process regarding the selection of the right chart for query results with one, two variables or three variable, one of them time related.

Graphs

Graphs are used for all the query results that are in the form of triplets and do not contain any time information. The presence of time is evaluated based on the OWL-Time [304] ontology of temporal concepts which has been created for describing the temporal properties of resources and is associated with the `http://www.w3.org/2006/time\#` namespace.

Graphs are dependent on many visualization parameters that can be adjusted to accommodate a wide range of query results including the layout algorithm that can be exploited to support the exploration of the results. Specific characteristics of the data to be visualized, such as the node size, node degree and the overall size directly affect choices regarding the overlapping percentage, the compactness and the edge length of the visualization. In order to specify the graph visualization parameters, the features of the query result are examined and evaluated based on the information available at the knowledge database. The evaluation method is presented here:

- *Node size. In order to determine the size of the nodes, the average string length of the labels available in the query result are examined. The value is then compared with the information available at the knowledge database,the minimum and maximum values of the label lengths of the endpoint. Based on this comparison, the node size is determined.*

- *Node degree. Aiming to determine the density of the query result, the average node degree is calculated. The value is then compared with the information available at the knowledge database, the average value of the node degree for the endpoint, and the result density is determined.*

- *Result size. The result size is compared to the maximum result size that the endpoint is support. In the case that this is reached, an additional parameter is added to the provided response, to ensure that the user is notified accordingly.*

- *Edge length. Nodes that do not contain long textual descriptions are relatively small in size and can be brought closer in the two-dimensional space. The maximum edge length is proportional to the node size.*

- *Node overlaps. The selected node size determines the allowed percentage of overlapping between nodes. As a rule of thumb, for larger nodes a higher degree of overlapping is acceptable given that the user gets an understanding of the information provided even without reading the complete information. Based on this rule the overlapping percentage is proportional of the node size.*

- *Edge crossing. Minimizing the number of edge crossings is very important for the readability of the graph, the path navigation and the node exploration. To accommodate that, special graph layouts are investigated and hierarchical and tree-like graph layouts are prioritized. Also, the default graph layout algorithm used if the query result does not comply to any specific structure, which is described in detail below, is designed to minimize the edge crossing.*

- *Graph area. Minimizing the overall graph area is not considered priority for the visualization. This is mainly due to the fact that the user interface offers many functionalities that support the exploration and navigation of the information, ensuring a user-friendly experience even when the graph is more than a few screens in dimensions.*

- *Node distribution. Uniform spatial distribution of the nodes ensures that the graph is easily explored and navigated. To this end, the minimum area that the graph covers is calculated proportionally to the result and node size.*

In the cases where the decision model has to provide visualization parameters for a SPARQL endpoint that has not been examined before, meaning that it is not included in the knowledge database, then the average values of all the endpoints included in the knowledge databases are used instead. The url of the endpoint is saved on the server and the endpoint is examined offline and included in the knowledge database if possible.

As already discussed, identifying the most suitable graph layout algorithm for the query result is key for the proper exploration of the information. Tree, hierarchical, star and circuit graph structures are first examined and applied if possible, in any other case the more generic solution of the force-directed layout algorithm is chosen. Also, for query result that contain symmetries, it is very important to respect and visualize them to ensure the comprehension of the visualized information. In details:

- *Tree or Hierarchical structure. In order to check if a directed graph, like a query result, is a tree we need to examine all triplets of information as follows. The first step is to locate the root node of the tree, a vertex with only outgoing edges. In the cases where there is more than one vertex with only outgoing edges or there is no such vertex then the triplets do not comply with the tree structure. After locating the root node we do a Depth First Search starting from it, if the search encounters the same vertex twice, indicating that it can be reached from two different paths, then the result does not comply with the tree structure. If the search concludes with unexplored vertices, then the graph is not connected and cannot therefore be a tree. If the Depth First Search* *contains all the nodes only once then the triplets comply with the tree structure and can be presented as such.*

- *Star layout. In order to identify a result that complies with the star layout we examine the node degrees. If the minimum node degree equals* 1*, the maximum node degree equals with* $resultsize - 1$ *and the average node degree equals* $2 * (resultsize - 1)/resultsize$ *then the query result has one central node and is presented using the star layout.*

- *Circuit layout. Similarly, the query result can be represented using the circuit layout when the minimum, maximum and average node degree equals with* 2*.*

- *Planar graph. Planar are called graphs that can be presented in the two-dimensional space without any edge crossing. We are not examining the possibility of offering the query result using a planar visualization, taking into account that there are many discussions about whether they offer any visual improvement over the non-planar visualization and the cost of computing if a graph is planar or not.*

- *Force-directed graph layout. For query results that does not comply with any of the above mentioned structures, we employ a generic yet robust algorithm for the graph layout. Force-directed graph drawing algorithms position the nodes of a graph in two-dimensional space trying to minimize edge length and edge crossing. The idea is to assign forces among the edges and the nodes and use them to minimize their energy. On one hand, spring-like attractive forces based on Hooke's law are used between nodes that are connected with an edge to bring the pair closer in space. On the other hand, repulsive forces like those of electrically charged particles are used between all pairs of nodes, no matter if they are connected or not. The balanced state of these forces, ensure that the edges have uniform lengths and nodes that are not connected are further apart in space.*

  Such algorithms have many benefits, as a result of the force balancing that ensures specific features. To begin with, the result is of high quality, offering uniform edge length, enforcing the spatial distribution of the nodes and emphasizing symmetries in the data. Also, it provides the needed flexibility as the balanced state of the forces can easily parameterized, as an example increasing the attractive forces brings the nodes closer in the two-dimensional space, based on the specific characteristics of the query result. Finally, the algorithm mimics the physical world providing an output that can be intuitively understood and accepted.

For example, for query results where the average string length is near the maximum that we have encountered the potential issue it that few nodes containing descriptive information will affect the overall dimensions of the nodes, expanding the space occupied by the graph and making its exploration a challenge. So, in such cases, the allowed overlapping percentage is 30%, the maximum allowed one and the node size is 150px, again the maximum one. This allows us to show to the user the majority of the information in the nodes, but also keep the overall size of the graph relatively small to allow the exploration of the connections between nodes. Similarly, the query result for all the synonyms of the term Author has result size four and a

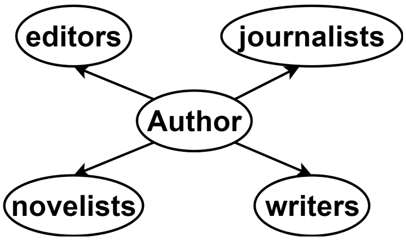


Figure 4.3: Star graphical representation of the synonyms for the word 'Author'

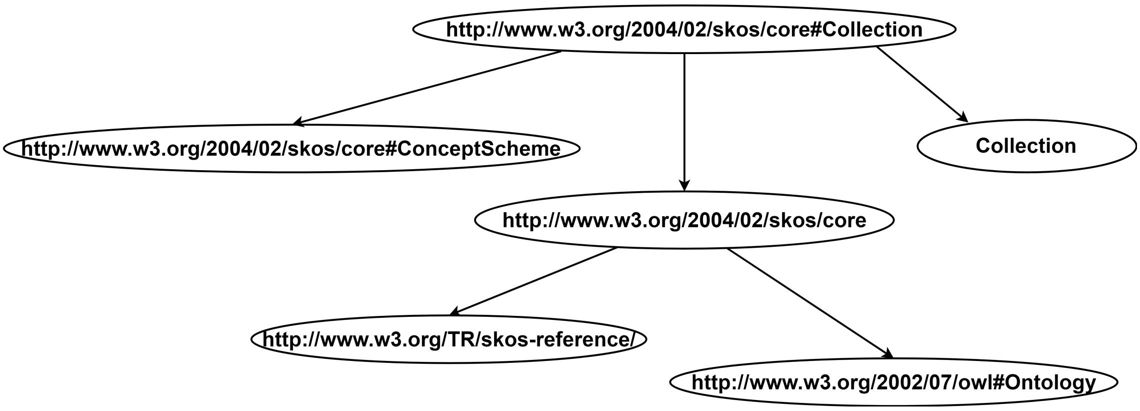

http://www.w3.org/2004/02/skos/core#Collection
http://www.w3.org/2004/02/skos/core#ConceptScheme
Collection
http://www.w3.org/2004/02/skos/core
http://www.w3.org/TR/skos-reference/
http://www.w3.org/2002/07/owl#Ontology

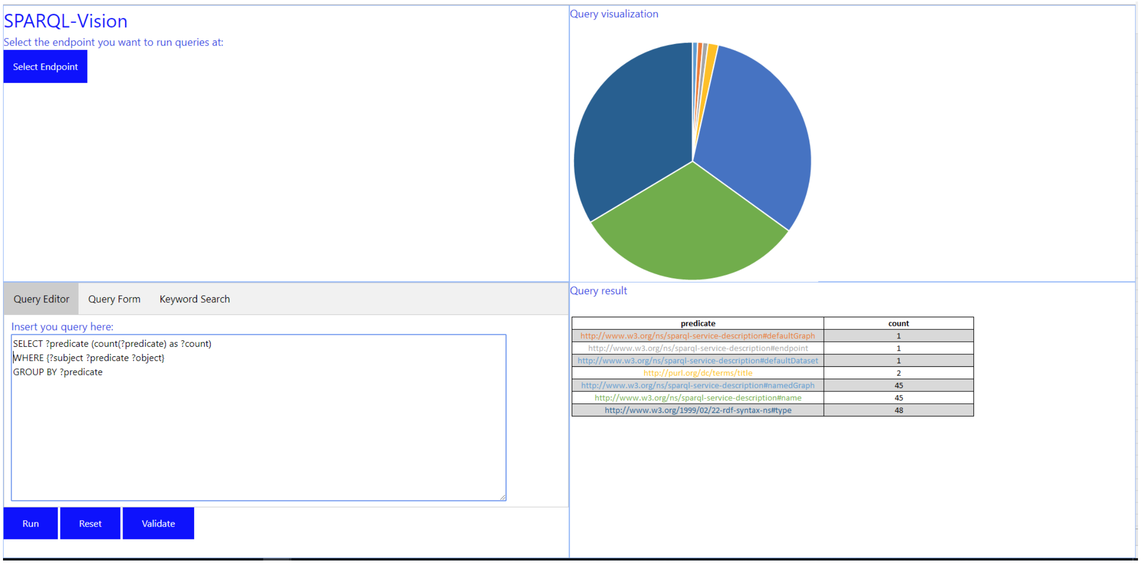
Figure 4.4: Tree-like representation of an exploration query

max node degree also four. This means that this query result should be visualized using the star layout algorithm as shown in Figure 4.3. In Figure 4.4 we present the exploratory query result for the term http://www.w3.org/2004/02/skos/core#Collection, which has been visualized using a tree-like approach due to the fact that there was one node with no incoming edges and no closed paths.

Charts

Choosing the right chart for the right data is not a scientifically defined task, it is based on intuition and takes into account the purpose of the visualization and the audience that the visualization is addressing. Aiming to provided a widely accepted set of rules, there are many studies that performed experiments aiming to identify the most suitable visualization type to be used based on the data format and content [308, 340, 88, 165, 178]. These studies have formed a set of empirical data visualization guidelines for choosing the right chart for the right data that are widely accepted. We have adopted these guidelines and we have extended them and matched them with query types in order to create specific decision rules for the model. We present here the defined rules, giving an intuitive description of the data visualization guideline, emphasizing the features of the query results that are important for each rule and concluding with the query types associated with each visualization type.

Pie chart. *We present first the pie chart as it is the only chart available that can represent only one variable. Specifically, it is used to represent the distribution of the 100% of a value, when there is no time element and there are at most 10 parts. If there are too many parts, or the percentage distribution is unbalanced, having one category over 90% and many others sharing the remaining 10%, then the visualization is not aesthetically appealing it is not displayed to the user. This visualization type is ideal for queries that have an aggregation with a group by clause*



```
SELECT ?predicate (count(?predicate) as ?count)
WHERE {?subject ?predicate ?object}
GROUP BY ?predicate
```

predicate	count
http://www.w3.org/ns/sparql-service-description#defaultGraph	1
http://www.w3.org/ns/sparql-service-description#endpoint	1
http://www.w3.org/ns/sparql-service-description#defaultDataset	1
http://purl.org/dc/terms/title	2
http://www.w3.org/ns/sparql-service-description#namedGraph	45
http://www.w3.org/ns/sparql-service-description#name	45
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	48

Figure 4.5: Count the times each predicate appears

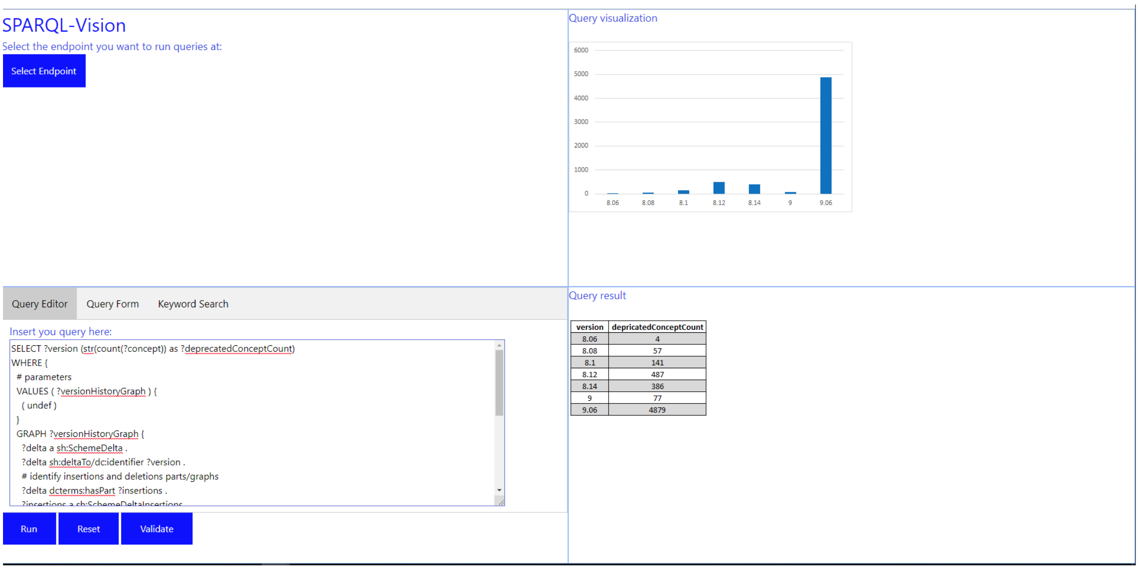
over a variable without any filtering, such as a having clause. The visualization may include, based on the user selection, also the labels of the categories represented by the percentages. This additional information is included in the textual representation where tuples are color matched with the chart.

Example 4.2.2 *As an example, a user interested in the times each predicate appears at the STW Thesaurus dataset would expect the result to be visualized as a pie chart. This is due to the fact that the query result represents the distribution of all the predicates in the dataset, there is a limited number of predicates to be visualized, only seven, provided by a group by query with an aggregation without any filtering. The visualization of this query result is shown in Figure 4.5, where the user has also chosen for the predicate labels to be included in the result.*

Bar charts. *Bar charts serve two purposes, the monitoring of one variable over the course of time or the quantification of a value related with a classification. Bar charts are preferred for small datasets, at most with 12 parts, like the monthly total value on the income of a company. The decision model chooses this chart when the extracted features indicate that the query result has only two variables, one of them is a numerical variable and the other is either time-related or descriptive, and the size of the result is less than 12. In contrast with pie charts, bar charts are not aware of the total distribution of the variable that they represent or if it an aggregated value. As an example, a bar chart can visualize the average age of retirement for people in ten most stressful work fields.*

Example 4.2.3 *Taking advantage of the skos-history version store which has been created with different versions of STW Thesaurus for Economics we can create useful queries that show the changes between versions over time. As an example we can retrieve the count of deprecated concepts between versions. The result of this query has two variables, one is time-related and a size of 7, so it is represented as a bar chart as shown in Figure 4.6.*

Line charts. *Line charts are complimentary to the bar charts, as they are used to monitor the same datasets but when the size is more than 12. In detail, line charts*



```
SELECT ?version (str(count(?concept)) as ?deprecatedConceptCount)
WHERE {
 # parameters
 VALUES ( ?versionHistoryGraph ) {
  ( undef )
 }
 GRAPH ?versionHistoryGraph {
  ?delta a sh:SchemeDelta .
  ?delta sh:deltaTo/dc:identifier ?version .
  # identify insertions and deletions parts/graphs
  ?delta dcterms:hasPart ?insertions .
  ?insertions a sh:SchemeDeltaInsertions
```

version	depricatedConceptCount
8.06	4
8.08	57
8.1	141
8.12	487
8.14	386
9	77
9.06	4879

Figure 4.6: Count of deprecated concepts between STW Thesaurus versions

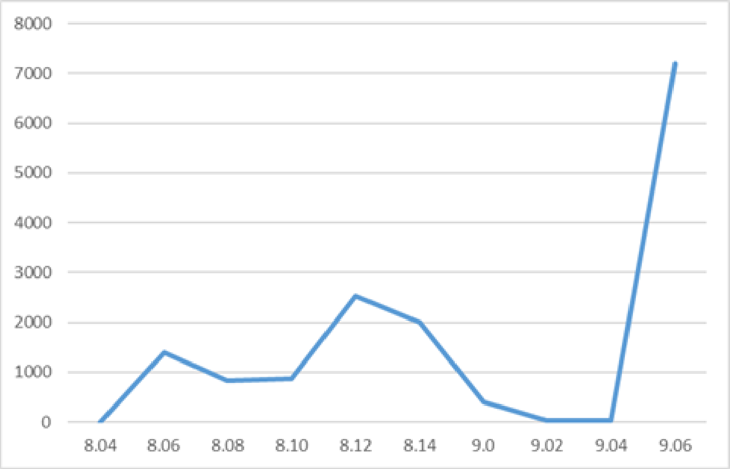


Figure 4.7: Count of new concepts inserted per STW Thesaurus version

show the variation of one variable with a lot of values over the course of time or the quantification of a value when classified over multiple classes. This visualization type is chosen when the query result has two variables with one of them numerical and the other either time related or descriptive and size more than 12.

Example 4.2.4 *Taking again advantage of the skos-history version store we can retrieve the count of new concepts inserted per version. The result of this query has two variables, one is time-related and a size of 10, so it is represented as a line charts as shown in Figure 4.7.*

Scatter-plot charts. *We have discussed the cases of datasets with two variables that include either a time element or descriptive information, we present here scatter-plot charts which are used for dataset that have two numerical values. They represent the information as spots over a grid-like area, where each axis can have a different scale to accommodate the minimum and maximum values of the representing variable. This chart does not have a size limit given that its overall dimensions are proportionally larger to the size of each visualized part. This visualization type* *is used for query results that contain exactly two numerical variables without a time element.*

Map charts. *Map charts are used to represent time invariable spatial information. This visualization type is used for datasets that contain spatial information, either coordinates or region names. The displayed information may be a numeric value, then a one-color visualization with gradient is used as the quantitative indicator, or descriptive where a multi-color schema for the categories is adopted. This visualization type is used for query results that contain spatial information. For uniformity reasons we categorize these queries as two-variable ones, one variable for representing the displayed information and one that provides the spatial element. Often, however, the spatial information may be available as two variable, latitude and longitude in the result. In such cases we merge the two variable in one pair of coordinates to facilitate their further processing.*

Area charts. *Area charts are used to represent how parts of a whole change over time. This visualization type is ideal for datasets that contain at least three variables, one of them is time related. The chart shows the variation of the variables and their comparison in each time stamp. It can be presented in two versions, the stacked area chart where the distribution of the variables is not examined and the 100% area chart where the variables represent the distribution of the 100% of a value at any given time. This chart is used for queries that have an aggregation along a group by clause over two variables, represented over time. Here, the presence of filtering functions is examined to ensure the use of the proper area chart type.*

Descriptive representation. *The chart types presented above do not cover all the possible cases for query results. In the case of a result with only one variable a chart is created only when the query contains a group bu clause with any filtering using a pie chart. In all the other cases the available information is not sufficient to create any visualization. Also, in the case where the query result has more than three attributes without any time element, the information is displayed in a structured table and no visualization is provided, given that the dataset in lacking the needed cohesion.*

We have implemented the decision model as a Java application with strict deterministic rules. The extracted features are mapped to the rules and the most fitting visualization type is selected.

4.2.3 DSS interface

We have implemented the DSS interface as a standalone Tomcat application that receives as input a JSON data object that contains the extracted features regarding the query result. The features that are included in the data object are such to support the choice of the visualization type independently of the content of the query, the endpoint queried or the underlying schema. The features included in the JSON object are:

- *Number & type of variables: list of variable types*
- *Time element: boolean*
- *Spatial element: boolean*
- *Result size: numerical*

- *Aggregation function: boolean*

- *Group by function: boolean*

- *Filtering function: boolean*

- *Average string length: numerical*

- *Min/Max/Average node degree: numerical*

- *Root node: boolean*

- *Depth First Search: list of nodes*

4.3 Integrated Platform

4.3.1 User Interface

A key component of the integrated platform is the user interface. This component is responsible for all the interactions with the user and allows the user to navigate and explore the information in an efficient and semantically meaningful way. It supports the composition of queries, ensures the validity of the user queries, presents to the user the retrieved information in dynamic and case-specific visualizations and allows the exploration of the information through filtering and isolation techniques. To achieve all the above a series of functionalities are provided through the user interface.

Query composition. *The user interface offers two ways for a user to write a SPARQL query. The first one is addressed to expert users, a single text box allows the user to type in any query. The second way is addressed to less experienced users that have only some basic knowledge and understanding of the SPARQL language. Here, the user is guided through the query composition by a form that has drop-down menus when applicable and requires minimal free text input from the user.*

Query validator. *To further support the user in composing their queries we have incorporated to the interface a query validator. After typing the query, the user can ask to validate the query and any issues are highlighted and the proper message with suggestions is shown. Even if the user chooses to submit the query without validating it, the validation process is always triggered before sending the query to the endpoint to avoid any invalid responses and error messages.*

Overview of the query result. *We present a text overview of the result to help the user get acquainted with it before exploring it. The textual representation of the query result is also used as a legend for the charts in case it is required by their type. This way the user can easily understand the displayed information by color matching the relevant text to the chart part. Furthermore, for the graph visualization, an additional overview is produced by isolating the most connected nodes, based on their node degrees, and visualizing them with respect to the spatial placement in the graph. The overview is interactive, meaning that the user can click on an area of the overview and the visualization panel is centered to the corresponding part of the graph.*

Dynamic visualization. *For large query results, or charts that are contain a lot of information such as scatter-plots, the mapping of the query result to the* *visualization part might not be obvious. For this reason, the query result visualization is not presented as a static image but as a dynamic object, responsive to user actions. This allows the user to select a part of the visualization and see highlighted the corresponding result part or chose a result part and see how it is depicted in the visualization.*

Customizable filtering functions. *As it will be presented in Section 4.4, only one in three endpoints enforces a query size limit and even in such cases it can be as high as* 100000*. Recognizing that navigating through such large query results can be ineffective and disengaging, the system provides to the user a series of filterng capabilities that can be used to restricted the displayed information. The interface enables the user to use one or multiple filtering criteria at the same time. A very characteristic example where this functionality is of great applicability is when the query contains multiple categories of semantic information. For example, when querying information about a specific terminology the result may contain information in different languages as well as synonyms and antonyms. A user can apply the filtering functions provided to display only results in English and terms that are synonyms to the original term. A different example would be a user visualizing aggregated the average expenses of a company per month. In this case, there might be a need to exclude from the visualization months before a limit. It is worth noting that all the provided filtering functions can also be achieved through an updated SPARQL query. However, by filtering the already retrieved information at the interface we allow the user to explore different filtering scenario in an interactive way, with a short response time and avoid over-querying the SPARQL endpoint.*

Interactive keyword search. *Another way designed to support the novice user in getting started with the exploration of a SPARQL endpoint and locate information of interest is through keyword search. The term is queried against the selected endpoint and the query results are presented to the user following the same data flow as any other result set.*

Path navigation. *For results that are visualized as graphs, the user is able to choose one node and follow all the paths of the graph that include this node, while the rest of the information is hidden. This way, irrelevant information is eliminated allowing the user to focus exclusively on the node and paths originating on it that are of interest. In contrast with most approaches in which a path can be traversed on one direction only, in our system when a node is chosen, all neighbors, either incoming or outgoing are visualized.*

Sub-result isolation. *The user can also choose to isolate a part of the result query, to better exploit and navigate the information that is present there. In order to make the isolation of the information user-friendly, the information to be isolated is chosen either by selecting the relevant part of the visualization or the text from the result overview.*

Exploration support. *Aiming to further support the novice user with the exploration of the available information the visualized result is interactive and upon user actions suggests further queries. An indicative example is when a user selects one specific result entity, then queries that retrieve its neighbors are suggested.*

4.3.2 Query processor

When a new query is submitted by the user a process is initiated by the integrated platform. First the query is validated, if any syntactic issues are identified, the user in notified and the query is not send to the endpoint until the user has corrected it. The valid queries are asked in real time to the selected endpoint. This approach is selected as it gives the integrated platform the flexibility to connect with any endpoint and provide to the user real time information based on the latest update. The alternative would be to duplicate the information of all the endpoints users are interested in to a dedicated back-end. This approach has main drawbacks as it is an expensive, time consuming process that would incapacitate the role of the SPARQL endpoints to provide up-to-day data. Also it would have a huge overhead to the back-end and threaten the sustainability of the platform as there would be need for storage of Terabytes of data and would delay the provision of information to the user in case he chose to visit an endpoint never before encountered by the platform. While temporary unavailability of the endpoint or poor performance may affect the user experience this is evaluated as a rarer and minimal discomfort to the user when compared with ensuring the flexibility and sustainability of our approach. The query result is then forwarded to the feature extractor.

4.3.3 Feature Extractor

The Extractor receives the query and the query result and extracts all the information needed for the DSS. This is properly formatted in a custom JSON object and forwarded to the DSS interface.

4.4 Experiments

In order to collect the raw data needed to create the knowledge database that is used by the decision model we have conducted a series of experimental analysis. The experiments were conducted using a laptop with an Intel(R) Core(TM) i7-4500U CPU at 1.80GHz, 4GB RAM memory connected to a 12Mbps home network connection.

Datasets. *In order to compile a knowledge database able to support our decision model we need to include information from a wide variety of SPARQL endpoints. We have examined over 140 SPARQL endpoints. The majority of the endpoints were not available during Autumn of 2019 when the testing was conducted or were available sporadically allowing us to obtain answers to only a few endpoints. We have managed to collect some measurements from* 55 *endpoints. Ten of them had strict query time limits and/or were very slow in providing responses so only few characteristics were collected. Another ten of the endpoints were not supporting queries using the strlength function, so we collected the rest characteristics. Forty five endpoints were available throughout the testing period and were responding to queries in a consistent and timely manner. For these endpoints the majority of the characteristics were collected.*

Metrics. *We collected the data needed for the knowledge database which are the result size limit, the total number of unique predicates, the total number of unique subjects and objects, the most/least connected subjects and objects, the size of the dataset, the most/least used predicates, the min/max string length for subject,objects* *and predicates. Based on the above information we have calculated and added to the knowledge database the average node degree.*

Methodology. *We have carefully chosen specific SPARQL queries that allow us to collect the needed information, for some queries we have created different versions in order to overcome the restrictions of some endpoints regarding the use of specific keywords, such as LIMIT and COUNT. We have created a Python script that accesses the endpoints, provided as a list of URLs, and runs the queries, recording the results in files as text. The script records a list with failed queries and tries to re-run them at a later time in case it is due to temporally unavailability of the endpoint. The results are then inspected and provided that there are no issues with the responses they are inserted into the relational knowledge database. The queries used for the collection of the information are presented here:*

- *Result size limit. SELECT ?subject ?predicate ?object WHERE {?subject ?predicate ?object}*
- *Number of unique predicates. SELECT (count(distinct ?predicate) as ?count) WHERE {?subject ?predicate ?object}*
- *Number of unique subjects. SELECT (count(distinct ?subject) as ?count) WHERE {?subject ?predicate ?object}*
- *Number of unique objects. SELECT (count(distinct ?object) as ?count) WHERE {?subject ?predicate ?object}*
- *Number of predicates. SELECT (count(?predicate) as ?count) WHERE {?subject ?predicate ?object}*
- *Minimum appearances of predicates. SELECT ?predicate (count(?predicate) as ?count) WHERE {?subject ?predicate ?object} GROUP BY ?predicate ORDER BY ASC(?count) LIMIT 1*
- *Minimum appearances of objects. SELECT ?object (count(?object) as ?count) WHERE {?subject ?predicate ?object} GROUP BY ?object ORDER BY ASC(?count) LIMIT 1*
- *Maximum appearances of predicates. SELECT ?predicate (count(?predicate) as ?count) WHERE {?subject ?predicate ?object} GROUP BY ?predicate ORDER BY DESC(?count) LIMIT 1*
- *Maximum appearances of objects. SELECT ?object (count(?object) as ?count) WHERE {?subject ?predicate ?object} GROUP BY ?object ORDER BY DESC(?count) LIMIT 1*
- *Minimum string length for predicates. SELECT ?predicate (strlen(str(?predicate)) as ?min) WHERE {?subject ?predicate ?object } ORDER BY ASC(?min) LIMIT 1*
- *Minimum string length for subjects. SELECT ?subject (strlen(str(?subject)) as ?min) WHERE {?subject ?predicate ?object } ORDER BY ASC(?min) LIMIT 1*

- *Minimum string length for objects. SELECT ?object (strlen(str(?object)) as ?min) WHERE {?subject ?predicate ?object } ORDER BY ASC(?min) LIMIT 1*

- *Maximum string length for predicates. SELECT ?predicate (strlen(str(?predicate)) as ?max) WHERE {?subject ?predicate ?object } ORDER BY desc(?max) LIMIT 1*

- *Maximum string length for subjects. SELECT ?subject (strlen(str(?subject)) as ?max) WHERE {?subject ?predicate ?object } ORDER BY desc(?max)*

- *Maximum string length for objects. SELECT ?object (strlen(str(?object)) as ?max) WHERE {?subject ?predicate ?object } ORDER BY desc(?max)*

Table 4.1: Experimental analysis of SPARQL endpoints

	Result size limit	Unique predicates	Unique subjects	Unique objects	Total predicates	Min subject degree	Min object degree	Max subject degree	Max object degree	Min predicate appearances	Max predicate appearances	Min strlen of predicate	Min strlen of subject	Min strlen of object	Max strlen of predicate	Max strlen of subject	Max strlen of object	Node Degree
Dbpedia [63]	10000	60649	2.4E+07	2.7E+07	4.4E+08	1	1	5	12856179	1	1.1E+08	47	28	32	47	404	51	0.12
URIBurner.com [316]	10000	75426	3.4E+07	3.8E+07	4.7E+08	1	1	426810	5696371	1	2.1E+07	35	2	1	56	76	576	0.15
OpenLink Virtuoso [238]	10000	4251	267167	347618	1533646	1	1	19748	90014	1	254487	3	1	1	121	331	23342	0.40
Allie Abbreviation Database [8]	10000	187	3.6E+07	3.7E+07	2E+08	1	1	184606	7642353	1	1E+08	29	12	1	76	49	4664	0.37
El Viajero's tourism [81]	10000	181	1019390	1127135	4631527	1	1	826	385491	1	1017691	29	12	1	75	358	65538	0.46
Lista de Encabezamientos de Materia [188]	30000	14	249328	537244	1757437	1	1	565	175758	29	351516	32	24	1	47	61	1651	0.45
OpenMobileNetwork [237]	40000	76	909837	1850866	2.3E+07	1	1	4847	357283	1	1.5E+07	30	30	1	73	200	671	0.12
Wiki Pathways [133]	100000	226	1139544	2012612	1.6E+07	1	1	4336	1230992	1	2028845	29	12	1	75	101	11239	0.19
BBC John Peel from DBTune [23]	None	25	76.255	122.136	349720	4	1	4	1	115	152301	47	30	1	47	71	2570	0.57
Magnatune from DBTune [195]	None	24	43.301	88.141	260487	16	15	16	15	265	43280	29	29	1	48	60	6468	0.50
STW Thesaurus for Economics [292]	None	7	48	99	143	2	1	48	45	1	48	30	2	2	62	55	58	1.03
Web-based Systems Group [147]	None	45	238	964	1662	1	1	31	79	2	233	30	72	1	63	158	6046	0.72
Alpine Ski Racers of Austria [9]	None	117	1267	1950	13441	1	1	145	1392	1	2745	29	32	32	74	218	1276	0.24
Datos [62]	None	32	677605	1936753	1.2E+07	1	1	18	322046	5	3000	36	49	63	55	268	282	0.22
Linked Open Vocabularies [187]	None	1269	178136	343748	861612	1	1	691	39154	1	157336	10	32	32	90	166	10538	0.61
Bio2RDF [32]	None	1	1.5E+07	144	1.4E+09	1	1	5750	125615	1	1.9E+08	39	29	1	66	65	93732	0.01
Open Data Thesaurus [236]	None	80	308	1351	3453	1	1	56	303	1	412	29	18	18	65	89	2617	0.48
OxPoints [241]	None	336	126315	313368	915052	1	1	21880	22978	1	126757	23	2	2	77	184	203920	0.48
Social Semantic Web Thesaurus [282]	None	159	12564	17844	127899	1	1	253	8879	1	14826	29	71	72	68	95	4937	0.24
Vacancies [317]	None	336	126315	313368	915052	1	1	21880	22978	1	126757	23	2	2	77	184	203920	0.48
Jamendo [151]	None	26	335.951	440.686	1385598	2	1	2	1	485	626242	29	29	1	60	94	33662	0.56
Geological Survey of Austria [109]	None	76	628	2936	7311	1	1	42	527	1	566	29	32	32	65	101	570	0.49
Isidore [150]	None	269	1.7E+07	70956381	4E+08	1	1	693397	26518	1	1.6E+08	29	12	1735	47	86	16492	0.22
DrugBank [76]	None	196	316950	1759602	3672531	4	1	568		2	251571	35	30	1	61	54	361	0.57
Revyu [256]	None	19	11105	21791	38359	1	1			1650	13005	47	35	1	63	81	790	0.86
UniProt [314]	None	214	1.2E+10	1.2E+10	5.5E+10	1	1			142047	9.5E+07	8	37	39	33	80	52	0.43
EventMedia [87]	10000	173	1.1E+07	5447856	1.1E+08	1	1	119	698903	1	2.9E+07			1	73		6	0.16
Camera dei deputati [186]	10000	357			2.4E+08	1	1		2204637	1	4.1E+07	37	44	4	43	69	46886	

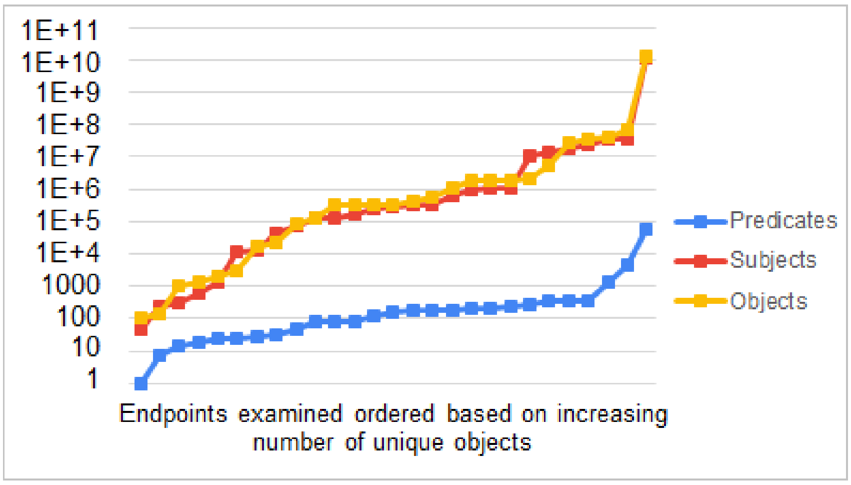


Figure 4.8: The number of unique subjects, predicates and objects for the examined endpoints

Results. *We present in Table 4.1 the results of the experimental analysis for some of the endpoints that we have evaluated. Aiming to keep the table readable and coherent we have added only the endpoints that have provided answers to all but one or two queries. Based on the results obtained we present here an in-depth analysis of the characteristics.*

- *Result size limit. From the* 45 *endpoints examined only* 15 *of them had any limit at the result size. The most popular limit was* 10.000 *but the value had a great deviation, from* 500 *to* 100.000 *elements and an average value of* 18.700*. This is a very important find as it sets the requirements regarding the volume of information that the user interface should be able to handle.*

- *Number of unique predicates. Endpoints have from* 1 *to* 75.426 *unique predicates, with an average value of* 4.388*. This deviation is very interesting with regard to the semantic differentiation of the datasets. Only a few datasets are focused on few semantic relationships between their entities, while most of them offer a higher variance.*

- *Number of unique subjects. The overall size of the dataset affects the number of unique subjects. The examined endpoints have from* 48 *to* 11.520.028.275 *unique subjects with an average value of* 343.734.281*.*

- *Number of unique objects. The overall size of the dataset affects also the number of unique objects. The examined endpoints have from* 99 *to* 12.153.725.295 *unique subjects with an average value of* 295.390.891*. While the minimum and maximum values are higher than the ones for the subjects, the average number of objects per endpoint is significantly lower from the average number of subjects. This is mostly due to the fact that few datasets contain descriptive information and free text while most of them depend on categories from the Semantic Web to describe the contained information.*
  In Figure 4.8 we present the chart with all the values of unique subjects, predicates and objects for the examined endpoints.

- *Number of predicates. The total number of predicates in a dataset represents the number of all the triplets of information and as a result the size of the*

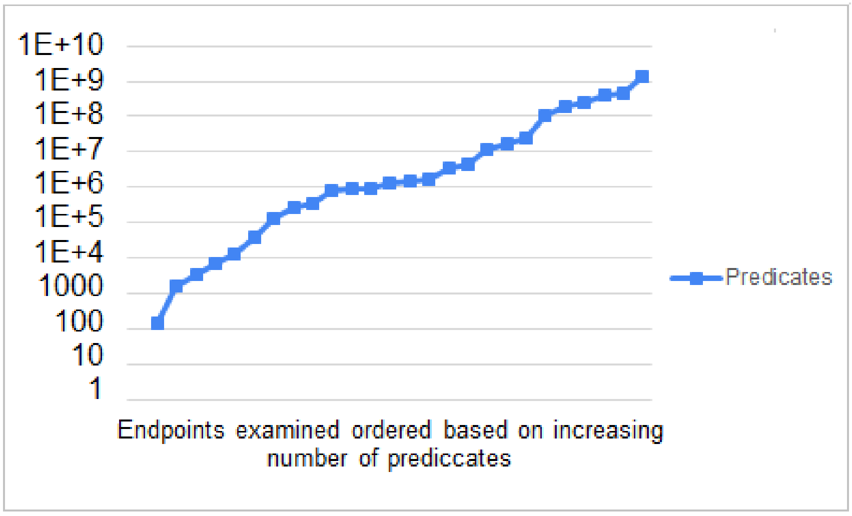


Figure 4.9: The number of predicates for the examined endpoints

dataset. Endpoints are dedicated to dataset ranging from 143 *to* 55.343.596.553 *triplets, and an average value of* 1.532.244.807*. The difference between the minimum and maximum values is indicative of the diversity of the datasets available through endpoints. The results are presented in Figure 4.9.*

- *Minimum & Maximum appearances of predicates. The minimum appearances of a predicate range from* 1 *to* 142.047 *depending on the type of dataset while the maximum times a predicate is repeated in a dataset can be as high as* 186.915.393 *for an endpoint. These value show that endpoints provide access to diverse datasets, some are focused on specific relationships between entities, thus having few predicates that repeat many times, while others cover a wide range of topics and concepts, limiting the re-usability of terms.*

- *Minimum & Maximum appearances of subjects. The minimum appearances of a subject range from* 1 *to* 16 *while the maximum can be as high as* 693.397 *for an endpoint. As expected, here too the diversity of the datasets is affecting the deviation of the values.*

- *Minimum & Maximum appearances of objects. For the objects this deviation is even more pronounced. Here, the minimum appearances of an object range from* 1 *to* 15 *while the maximum can be as high as* 12.856.179 *for an endpoint. Similarly with the predicates, datasets that are providing information for specific scientific fields repeat key concepts and semantic terms multiple times. In Figure 4.10 we present the chart with all the values of appearances for subjects, predicates and objects for the examined endpoints.*

- *Minimum & Maximum string length for predicates. The string length for the predicates ranges from* 47 *to* 371 *characters, with an average value of* 62*. These values are within the expected range as predicates are mostly URLs from the Semantic Web.*

- *Minimum & Maximum string length for subjects. The string length for the subjects ranges from* 72 *to* 404 *characters, with an average value of* 87*. These*

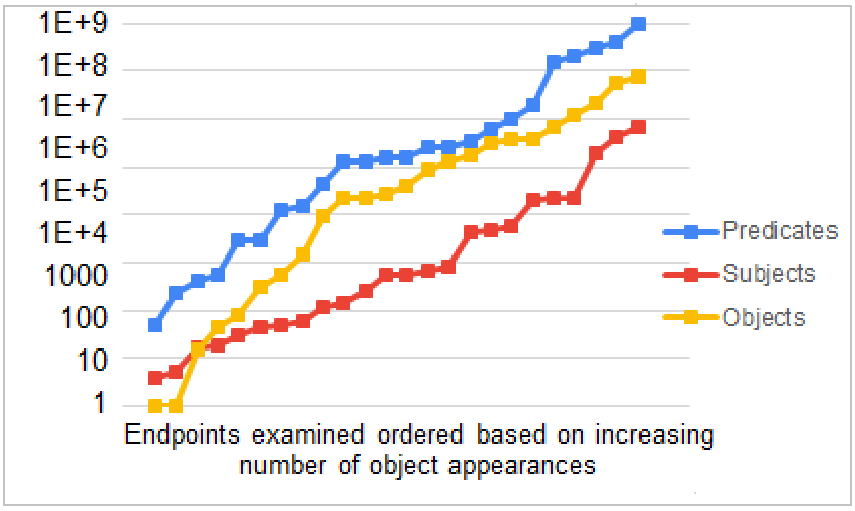


Figure 4.10: The number of appearances for subjects, predicates and objects for the examined endpoints

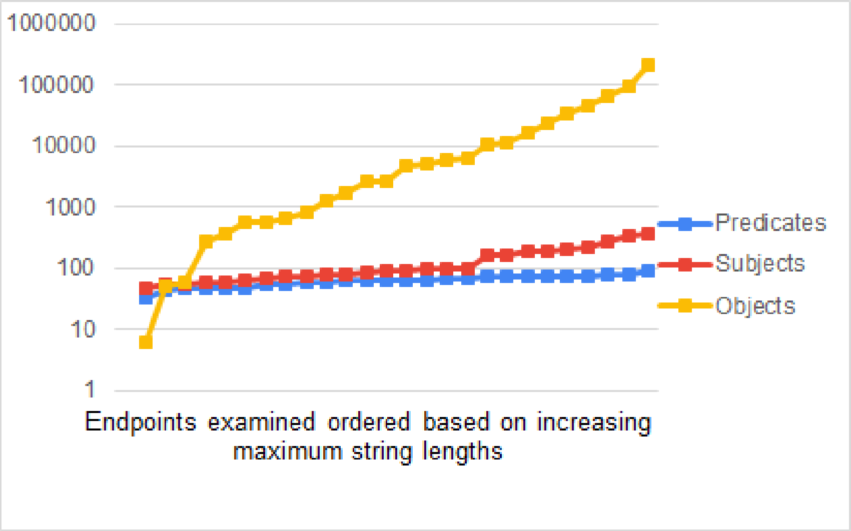


Figure 4.11: The maximum string lengths for subjects, predicates and objects for the examined endpoints

values are higher than the ones for the predicates as subjects sporadically include free text in addition to URLs from the Semantic Web.

- *Minimum & Maximum string length for objects. The string length for the objects ranges from* 72 *to* 203.920 *characters, with an average value of* 36.262. *Given that objects are mostly descriptive and include a lot of free text these values are again expected.*

  In Figure 4.11 we present the chart with the maximum string lengths for subjects, predicates and objects for the examined endpoints and in Figure 4.12 the minimum string lengths.

- *Average node degree. The average node degree for the datasets is* 0.32, *indicating that the datasets are not very connected and probably include independent sub-graphs. The distribution of the values is shown in Figure 4.13.*

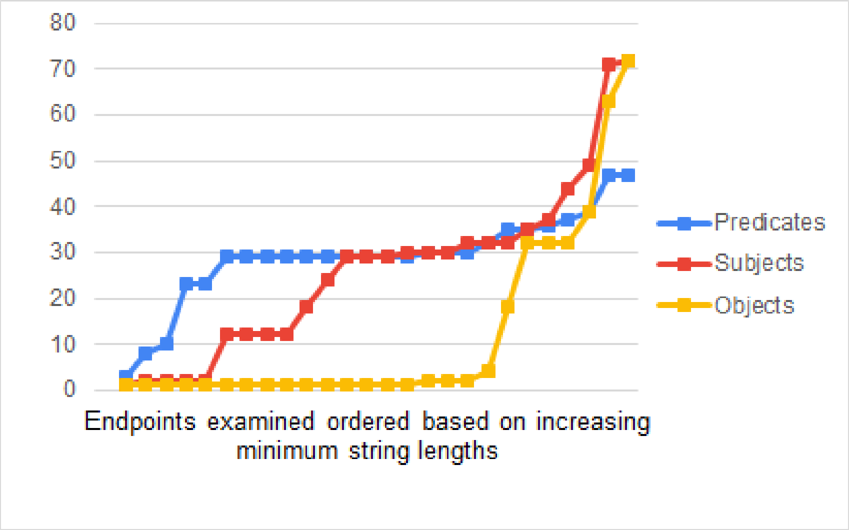


Figure 4.12: The minimum string lengths for subjects, predicates and objects for the examined endpoints

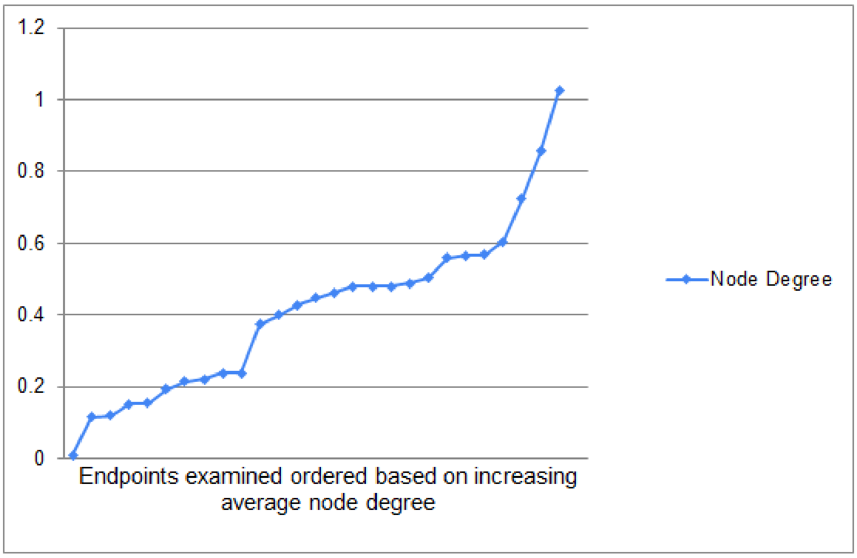


Figure 4.13: The node degree for the examined endpoints

Conclusions

In this chapter, a novel system architecture that supports both expert and novice users with querying SPARQL endpoints, exploring and visualizing the query results in a dynamic way was presented. The discussed system has been designed in a schema agnostic and data structure robust way that allows the visualization of diverse query results in a user-friendly and interactive way.

Chapter 5

Scalable exploration

In many cases, such as road maps, communication networks, biological structures, financial and blockchain transactions, the datasets are complicated, highly connected, contain critical information that should be available complete and need to be accessible by many users with different analysis needs. In these cases, the challenges of scalability, interactive and consistent visualization without loss of information and data exploration through different filtering and aggregation functions should be met.

In order to achieve that, a novel technique for the pre-processing, indexing and storage as well as visualization of very large datasets is proposed. The technique has been carefully designed to alleviate the restrictions of the above-mentioned approaches on the input datasets, regarding accompanying metadata related to the model, full or partial semantic annotation, hierarchical structure, data completeness and size limitation. The technique is based on a modular off-line pre-processing phase that allows us to meet all the requirements.

Independence of the characteristics of the input dataset. *No part of the pre-processing phase is based on specific data formats allowing us to handle any dataset including incomplete datasets, datasets that are not annotated with respect to semantic categories or datasets that do not comply with the hierarchical model.*

Scalability with regard to the size and node degree of the input dataset. *Our technique allows us to handle real datasets, with variable volume and connectivity degree. To achieve this, we split the input dataset into smaller partitions and visualize them as independent graphs. We propose two different algorithms for the arrangement of the visualized partitions into the two-dimensional Euclidean space that minimize the length of the external edges based on novel cost functions. This way we achieve smaller navigation paths, consistent exploration and limit the visualization noise efficiently for millions of nodes.*

Innovative storage schema that ensures efficient exploration of the information. *The pre-processing phase results in all the elements of the input dataset to be assigned coordinates with respect to the Euclidean space allowing the storage of the dataset in a structured way to a spatial database.*

Dedicated spatial indexing. *We index the spatial information in a way that ensures the reliable and efficient retrieval of information. This allows the filtering of the data and the creation of multiple abstraction layers using various criteria that can be custom to the input dataset. In addition context search is supported through dedicated indexing.*

Fully fledged prototype system. *In order to showcase the usability of our tech-*

nique we have implemented a dedicated API over the storage schema along with a client-server architecture that allows the exploration and the navigation of the information through a web UI. The system offers an interactive overview of the visualized information, allowing the user to choose intuitively the area to further explore, interactive visualization over different zoom and abstraction layers and keyword search.

The client-server communication is based on the translation of the user operations through the UI into simple and very efficient spatial operations over the stored data, allowing the unobstructive and smooth exploration of the information without any dependencies on the computational resources of the user device. The Interactive Visualization of Very Large Graphs (IVLG) prototype system is publicly available at [148].

Experimental analysis. *We performed a thorough experimental study on our technique. We employed our technique for the visualization of many real and synthetic datasets with as many as 150 million elements and an average node degree as high as 20, in order to show that we can handle large and diverse datasets without any limitations to their characteristics. In addition, we compare our system with four other ones with respect to the rendering time per data object. Our system outperforms the others by at least 60%.*

5.1 System Architecture

We present here the three phases of the off-line, pre-processing technique that allow us to efficiently represent large linked dataset as one continuous graph.

The first phase of the technique ensures that we are able to handle very large datasets. For this reason, the input dataset is split in smaller parts. The number and the size of these parts are decided based on the size, number of entities and relationships, and characteristics, connectivity degree and average label length, of the input dataset. Connections between the parts of the input dataset are separately stored to be introduced again in the third phase of the technique.

The second phase of the technique is dedicated to the proper representation of the parts of the input dataset. Each part is transformed into an independent graph, using the Scalable Force Directed Placement algorithm [158]. We chose this algorithm due to its flexibility and robustness.

We carefully initialize the parameters of the algorithm based on the characteristics of the parts. Our aim is to achieve the right degree of compactness, avoid any overlapping and ensure that each sub-graph has approximately the same horizontal and vertical span. This is of high importance to ensure that the merged information will have a uniform spatial distribution and will be explorable.

In the third phase of the technique, the sub-graphs are merged into one continuous graph. This means that each entity of the input dataset is transformed into a graph node that has a unique position in the two-dimensional space.

It is important to arrange the sub-graphs in a way that the connections between the sub-graphs, as identified and stored at the first phase of the technique, are introduced here as edges between the nodes in a way that minimizes their total length. For this reason, we have implemented a heuristic algorithm, which uses a cost function based on the length of the edges, to arrange the sub-graphs.

Finally, the graphical information is translated to a spatial storage schema and made accessible through a flexible, complete and efficient API.

5.1.1 Phase A: Dataset partitioning

This phase of the proposed technique is responsible for receiving as input the linked dataset, model the information with respect to the graph model and partition the graph into smaller sub-graphs. Linked datasets can be mapped to a graph $G = (\mathcal{V}, \mathcal{E})$ *where* $\mathcal{V}$ *is a set of nodes (v) that represent the entities of the input dataset and* $\mathcal{E}$ *is a set of edges (e) that represent the connections between the entities.*

Graph partitioning is an important problem that has extensive applications in many areas, including scientific computing, VLSI design, and task scheduling. The problem can be intuitively defined as the partitioning of the vertices of a graph in k roughly equal parts, in number of vertices, such that the number of edges connecting vertices in different parts is minimized.

Formally, given a graph $G = (\mathcal{V}, \mathcal{E})$ *with* $|\mathcal{V}| = n$*, the k-way graph partitioning is defined as the partitioning of* $\mathcal{V}$ *into k subsets,* $\mathcal{V}_1, \mathcal{V}_2, ..., \mathcal{V}_k$ *where* $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset$ *for* $i \neq j$*,* $|\mathcal{V}_i| \approx n/k$*, and* $\cup_i \mathcal{V}_i = \mathcal{V}$*.*

The k-way graph partitioning aims to minimize the number of edges of $\mathcal{E}$*, the vertices of which belong to different subsets. Given a graph* $G = (\mathcal{V}, \mathcal{E})$ *and a k-way graph partitioning, the edges of the graph G, the vertices of which belong to different subsets, are called external edges.* $\mathcal{X}_{ij} \subset \mathcal{E}$ *is the set of external edges between two subsets* $\mathcal{V}_i$ *and* $\mathcal{V}_j$ *where* $i \neq j$ *and* $\mathcal{X}$ *is the set of all external edges between all possible pairs of* $\mathcal{V}_i \in \mathcal{V}$*.*

In order to ensure that all datasets, no matter their specific characteristics or volume, are partitioned properly, we chose a robust algorithm with the capability of handling datasets ranging from a few thousands to millions, and from sparse to dense. We use the k-way graph partitioning algorithm of G. Karypis and V. Kumar [161] as implemented in Metis [1]*, as it meets the requirements of handling any input dataset without any restrictions on its size and properties.*

Example 5.1.1 *For the 48S pre-initiation protein complex, that has been presented in Chapter 2, the k-way partitioning divides the dataset into nine sub-graphs, as shown in Figure 5.1. Given the 116774 atoms of the dataset each partition has 13.000 atoms. The number of the partitions is selected to support the second phase of the technique as presented in 5.1.2 and create a continuous graph with similar horizontal and vertical dimensions. Based on the small size of the input dataset here, the possible options are 4, 9 or 16 partitions. While with 4 partitions the sub-graphs have too many elements for the second phase, with 16 they have unnecessary few.*

5.1.2 Phase B: Graphical representation of the sub-graphs

This phase of our technique is dedicated to the transformation of each part of the input dataset to an independent sub-graph. The main requirements of this phase are two, both related with the spatial distribution of the nodes. To begin with, we need to achieve a uniform density of the nodes in the two- dimensional space for each sub-graph. This means that the nodes of the sub-graph should be spaced out enough not to have any overlaps between them while at the same time as close as possible in order not to have a lot of empty space.

[1] `http://glaros.dtc.umn.edu/gkhome/metis/metis/overview`

Table 5.1: Number of external edges between the nine CNs

	CN 1	**CN 2**	**CN 3**	**CN 4**	**CN 5**	**CN 6**	**CN 7**	**CN 8**	**CN 9**
CN 1	0	25	16	27	3	24	1	26	6
CN 2	25	0	17	24	4	6	14	25	11
CN 3	16	17	0	10	25	17	19	4	1
CN 4	27	24	10	0	26	18	13	12	7
CN 5	3	4	25	26	0	14	27	21	11
CN 6	24	6	17	18	14	0	14	5	11
CN 7	1	14	19	13	27	14	0	10	16
CN 8	26	25	4	12	21	5	10	0	9
CN 9	6	11	1	7	11	11	16	9	0

Moreover, we should have sub-graphs requiring approximately the same horizontal and vertical space. This is very important for the next phase of the proposed technique, which connects all the sub-graphs into one continuous graph by arranging them in a two-dimensional grid. Given that the dimensions of each grid cell are based on the width and height of the included sub-graph, having sub-graphs with similar sizes ensures the lack of empty spaces.

Both requirements should be achieved independently of the specific characteristics of the sub-graphs. Given the expected diversity of the input datasets, the sub-graphs are also expected to have different characteristics, ranging from sparse to dense and with nodes with uniform sizes, when entities are identifiers, to nodes that differ a lot, when entities contain descriptive information.

For this reason, we choose the Scalable Force Directed Placement algorithm [158] for the graphical representation of the sub-graphs. The algorithm is based on a physics approach, balancing a system where nodes are considered charged particles and edges are modeled as springs. This allows for the parameterization of many attributes for the output graph including the allowed overlapping percentage, the size of the nodes and the size of the graph.

5.1.3 Phase C: Merging the sub-graphs into one continuous graph

In this section, the third phase of the technique that connects the sub-graphs in one continuous graph is described. Given that at we dynamically choose the number of sub-graphs based on the input dataset, we need a solution for the sub-graph arrangement that accepts as input any number of partitions and any set of external edges between them and results to an arrangement of the sub-graphs in the two-dimensional space in a way that the length of long edges connecting different sub-graphs is minimized.

Problem Definition. *Let* $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_k\}$ *where each* $\mathcal{P}_i$ *for* $i = 1, 2, ..., k$ *corresponds to the i-th partition and includes the vertices of the* $\mathcal{V}_i$ *sub-graph with their coordinates and the edges between them. For each partition* $\mathcal{P}_i$ *a* CN_i, *a Complex Node (CN) is defined. A Complex Node* CN_i *refers to the two-dimensional area with dimensions the width and height of the* $\mathcal{P}_i$ *and content all the graph elements, nodes and edges, of the partition. The position of each graph element in the continuous graph is relevant to the position of the Complex Node that contains it to the two-dimensional space.*

Problem Transformation. *Let CN denote the set of all Complex Nodes for the partitions, with* $|\mathcal{CN}| = k$ *and* $\mathcal{W}$ *the set containing the weights between the Complex Nodes as defined by the number of external edges between them,* $W(i, j) = X_{i,j}$. *The*

Table 5.2: Steps of executing the proposed algorithm for the nine CNs

Step 1	CN 4 is set at the center of the grid
Step 2	CNs are ranked based on the weight [1,5,2,6,7,8,3,9]
Step 3	CN 1 is placed next in the grid
Step 4	CNs are ranked based on the weight [2,6,8,5,3,7,9]
Step 5	CN 2 is placed next in the grid
Step 6	CNs are ranked based on the weight [8,6,3,5,7,9]
Step 7	CN 8 is placed next in the grid
Step 8	CNs are ranked based on the weight [3,6,5,7,9]
Step 9	CN 3 is placed next in the grid
Step 10	CNs are ranked based on the weight [5,6,7,9]
Step 11	CN 5 is placed next in the grid
Step 12	CNs are ranked based on the weight [6,7,9]
Step 13	CN 6 is placed next in the grid
Step 14	CNs are ranked based on the weight [7,9]
Step 15	CN 7 is placed next in the grid
Step 16	CN 9 is placed next in the grid

process of placing each $CN_i \in \mathcal{CN}$ into the two-dimensional Cartesian integer grid by assigning grid positions to each CN_i, such that it minimizes the weights is defined as Grid Arrangement Problem (GAP). The problem is proven [239] to be NP-hard for every dimension of the grid.

Problem Solution. *We present here a greedy heuristic algorithm, as a solution for the Grid Arrangement Problem that can be applies with large number of nodes. The algorithm uses a weight-cost function to calculate the cost of adding a CN to a specific position given a grid arrangement. The algorithm, incrementally creates the grid arrangement by selecting, in each iteration, the CN which is going to be to placed in the grid. The CN, with the highest cost as provided by the weight-cost function, is selected on each iteration. The weight-cost function is calculating the sum of the weights between each examined CN to the ones already placed on the grid.*

Example 5.1.2 *We present the weights between the CNs, of the human cap-dependent 48S pre-initiation complex, in Table 5.1. To simplify the example the numbers provided in the table have been scaled down, keeping the initial ratio. First, we merge the sub-graphs into one continuous graph, by following the steps of the proposed algorithm. We present the steps followed as well as the intermediate results of the cost function in 5.2. Then, we merge the sub-graphs into one continuous graph in a random way. The results of the two placements is presented in Figure 5.1. There is a 40% decrease to the overall weight cost.*

The merged information is stored in a distributed, scalable Accumulo [2] key/value datastore where the GeoMesa [3] XZ-ordering index is used for the spatial information of the graph.

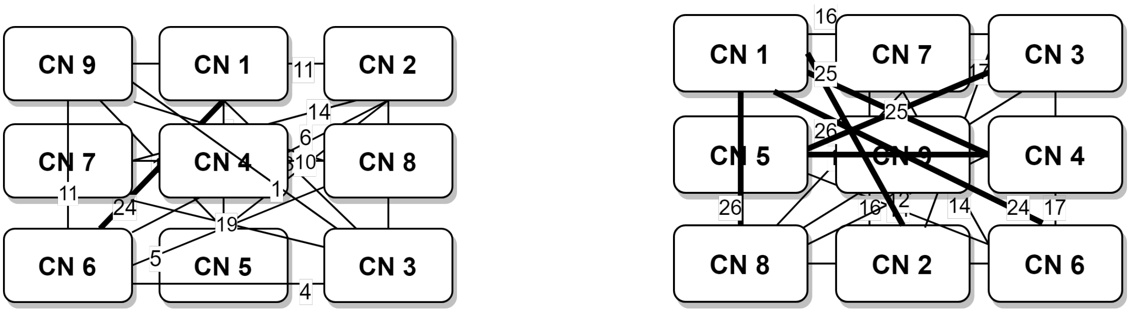
5.1.4 Use cases

In order to provide access to the graph of the input dataset in an efficient and user-friendly way, we have implemented a dedicated API that allows the retrieval of the information in different ways, including spatial and filtering queries and keyword search.

We have implemented a standalone Tomcat application, written using Java servlets,that receives user requests along with multiple parameters and returns the result as an

[2] `https://accumulo.apache.org/`

[3] `https://www.geomesa.org/documentation/current/index.html`



(a) Placement based on the proposed algorithm

(b) Random placement

Figure 5.1: CN placement in the two-dimensional Cartesian integer grid

XML file, that follows the GraphML format and includes the proper geometry parameters that specify the position of the nodes in the two-dimensional space. The API is designed to receive any Common Query Language function, including all the spatial functions and geometries.

We are going to showcase the usability of the API through a series of use cases as identified in the presented use case in Chapter 2, regarding the exploration and study of the human cap-dependent 48S pre-initiation complex which represents 47 unique protein chains with 116774 atom count. These use case are the following:

Example 5.1.3 *Keyword Search. The initial exploration of the dataset begins when the user wants to locate information of interest. The intuitive way for this is to use a keyword and search among the entities of the dataset. For a protein complex the user may search the identifier of an atom or a group of atoms. The API will return a list containing all the entities containing the given keyword along with their spatial information.*

Example 5.1.4 *Spatial queries. After locating an entity of interest, the user can use its spatial information and query the API over the two-dimensional space, using any Common Query Language spatial function and geometry. As an example, a user interested in identifying specific connections in parts of the biological structure, responsible for a specific function, can perform spatial queries and retrieve this information.*

Example 5.1.5 *Filtering based on connection types. The API enables the retrieval of the information using one or multiple filtering criteria at the same time. The user can easily select to isolate one or more types of connections between atoms based on their semantic annotations, and can combine this filtering with a spatial query for a specific area of the graph.*

Example 5.1.6 *Path exploration. After locating an entity that is of interest, the user is able to retrieve all the paths of the graph that include this node, up to a customisable length. In contrast to most approaches in which a path can be traversed only with respect to the edge direction, in our API when a node is selected, all neighbors, either incoming or outgoing are retrieved.*

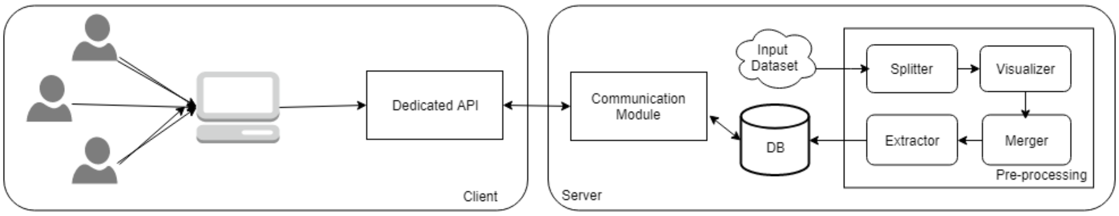


Figure 5.2: System Architecture

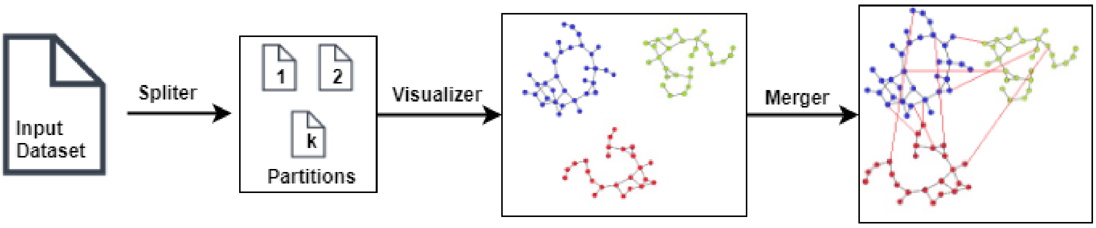


Figure 5.3: System Architecture

5.2 Integrated Platform

5.2.1 Platform Architecture

The architecture of the IVLG system is based on the server-client model as shown in Figure 5.2. The server side of the system is separated in 3 main modules, the offline module, where the input dataset is translated into one fully connected graph, the communication module which makes available the visualized information, and the storage module, which stores the spatial information of the graph.

The offline module is further separated into 4 sub-modules, each responsible for the necessary tasks that ensure the proper handling of any input dataset, its visualization with respect to the Euclidean space and its storage in the DB with a schema that allows the efficient retrieval of information. These modules are the following:

Splitter. *The input dataset is split into smaller partitions using the METIS [205] algorithm. This algorithm tries to find the optimal partitioning of the nodes in a way that minimizes the placement of connected nodes in different partitions. Given that linked datasets cannot be partitioned in independent sub-graphs, the information about the connections of nodes in different partitions, which is referred to as external edges, is kept to be used by the Merger sub-module.*

Visualizer. *Each partition is visualized as an independent graph. The visualization algorithm used depends on the input dataset and its characteristics or on specific requirements for the visualization, such as compactness, overlapping and orientation. Currently, IVLG uses the Scalable Force Directed Placement algorithm [158], as it has no limitation for the characteristics of the dataset and allows for the parameterization of many of the graph attributes including the allowed overlapping percentage, the size of the nodes and the size of the overall visualization.*

Merger. *This sub-module receives as input the visualized partitions and the information regarding the external edges between the partitions and produces one continuous graph as shown in Figure 5.3. The aim of this module is the arrangement of the partitions in the two-dimensional Euclidean space in a way that the length of long edges connecting them is minimized. This is of high importance for the quality* *of the produced visualization as placing two connected nodes close, minimizes the average navigation path, ensuring the meaningful navigation of the information even when accessing it through a device with a small screen.*

Also, minimizing the total length of the external edges, results in less background noise for the visualization caused by long edges. We map the problem of arranging the partitions in the two-dimensional space to the Grid Arrangement Problem for which we developed a novel cost function to estimate the length of external edges between partitions. This is necessary as calculating the actual length of the external edges between partitions for all possible arrangements is a NP-hard problem [239].

Extractor. *The Merger transforms the dataset into one continuous graph, on which abstraction and filtering functions are calculated. Each function indicates the criteria used to assign an abstraction score to each node, a number of abstraction levels and a threshold for each level. Each node is assigned a score, which is used to determine in how many of the abstraction levels is present. This calculation is done at the offline module to allow the application of complex criteria to datasets with millions of elements without affecting the overall system performance.*

The spatial information about the graph elements is handled by the storage module, where the triplets of the graph are stored in a distributed Geomesa-Accumulo database and indexed with a XZ-index. The client side consists of a user-friendly interface, which allows multiple users to navigate and explore the information simultaneously, though a laptop or tablet.

Part of the client side of the system is also the Translator, a module which depicts the user actions into spatial queries with respect to the Euclidean space before sending the request to the server. This module is the core client component that ensures that the user can navigate the graph in a meaningful way with respect to the stored spatial information regardless of any visualization preferences.

This module takes into consideration the size of the screen and the area of the graph that fits there along with the user's choices on navigation, panning or zooming, and on exploration using different filtering and aggregation criteria. This way when the user moves to a different part of the graph, regardless of the way that this action was triggered, the relevant movement, with respect to all the chosen visualization settings, is translated to the corresponding coordinates of the graph to the Euclidian space, before the request is sent to the server.

The Linker is responsible for receiving client requests, querying the DB for the required information and returning to the client the elements to be visualized gradually in small packages that can be handled without causing any performance issues.

5.2.2 Platform Functionalities

To allow the user to navigate and explore the information in a fast and semantically meaningful way, a series of functionalities are provided through the web interface. These take advantage of the storage schema at the server, the indexing and the dedicated API to perform complex queries over the dataset as they are answered in real time.

Overview of the input dataset. *Understanding the user needs to get acquainted with the dataset before exploring it, we present a general overview of the whole dataset as a static image. This image is produced by isolating the most connected nodes, based on their node degrees, and visualizing them with respect to the* *spacial information. The overview is used to locate areas of the graph that seem of interest. The user can click on that area on the static image and her screen is centred to the corresponding part of the graph on the interactive visualization panel. An additional feature that enables the user to navigate the dataset overview as an interactive visualization, is the zoom level that can be up to 0.1%.*

Interactive visualization. *The graph visualization is interactive and responsive to any user request, allowing the user not only to browse information at the same abstraction layer with simple panning and scrolling actions but also through different abstraction and zoom levels. This is achieved due to the Translator module, which can translate any change of the user's preferences, regarding the graph area currently visualized, to the corresponding spatial query to the backend.*

Customizable filtering and aggregation. *Recognizing that navigating through millions of nodes to locate the needed information is an ineffective and disengaging approach, the system enables the user to view the information using one or multiple filtering and aggregation criteria at the same time. The user is also able to define the abstraction level that allows her to focus on the information of interest.*

A very characteristic example where this functionality is of great applicability is when the input dataset is highly connected and at the same time very diverse, such as graphs about network traffic, shopping choices and social networks. Then the user might be interested in only one aspect of the information, while at the same time looking for nodes with specific node degree, a filtering combination that can be easily achieved.

Interactive keyword search. *Another way the user can locate the information is through keyword search. The term is matched against both node and edge labels, ensuring that no information will be overlooked. The search result is an interactive list of nodes containing the given keyword. By clicking on a search result, the navigation window is relocated on the position of the graph containing this node, while the node is highlighted by a bounding box.*

Path navigation. *After locating an area of the graph that is of interest, the user is able to choose one node and follow all the paths of the graph that include this node, while the rest of the information is hidden. This way, irrelevant information and background noise are eliminated allowing the user to focus exclusively on the node and paths originating on it that are of interest. In contrast with most approaches in which a path can be traversed on one direction only, in IVLG when a node is chosen, all neighbours, either incoming or outgoing are visualized.*

On the fly sub-graph isolation. *The user can also choose to isolate a part of the graph, to better exploit and navigate the information that is present there. In order to make the isolation of the information user-friendly, the information to be isolated is chosen by selecting individual nodes, search results or areas of the graph present in the current screen.*

As a further step the user can chose the length of the path from the chosen nodes that are included in the visualization. The isolated information is shown in an independent window allowing the user to choose in addition to any filtering and aggregation functions the algorithm that will be used for the visualization. Given that the information chosen here may be spread out, respecting the spatial information of the continuous graph would result in a chaotic and sparse visualization that could unnecessarily extend in many screens. To avoid this, the stored spatial information is not used here. The chosen nodes and their edges are visualized on the fly as one

Table 5.3: Synthetic Datasets

Dataset	**#Edges**	**#Nodes**	**Degree**
1M D5	5.1M	1M	5.09
10M D5	50.2M	10M	5.02
10M D10	103M	10M	10.3
10M D15	157M	10M	15.7
50M D5	259M	50M	5.18

graph independent from the rest dataset.

This functionality can easily support the on the fly visualization of 5.000 graph elements, while still providing all the system functionalities, such as the filtering and aggregation functions, the zoom and abstraction levels. This is done by taking advantage of the computational ability of the server side of the system. The optimal implementation of many visualization algorithms by the Graphviz [113] platform is also exploited allowing the user to view the isolated graph visualized by the algorithm of her choice. Once more the Linker is used to send to the user the visualized information gradually in small packages that can be easily handled by devices with limited computational resources.

5.3 Experimental analysis

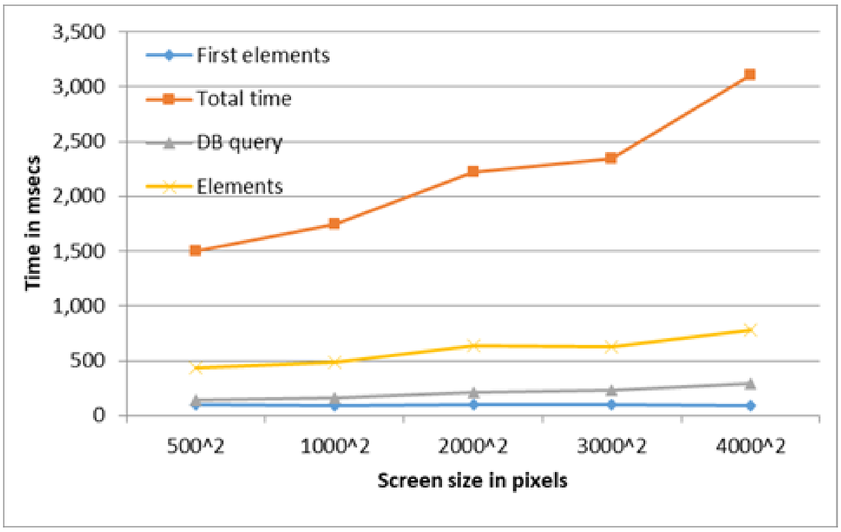
In order to evaluate the proposed technique and carry out an experimental analysis on the efficiency of the information retrieval through the API we developed a web tool. We used the tool to spatially query the API and evaluate the response time. The experiments were conducted using a laptop with an Intel(R) Core(TM) i7-4500U CPU at 1.80GHz, 4GB RAM memory over a 12Mbps network connection.

Datasets. *In order to test our system with many diverse datasets with a wide range of characteristics we used synthetic datasets. The synthetic datasets, presented in Table 5.3, were created to comply with specific requirements for number of nodes and node degree to showcase the scalability of the technique to the size and density as well as the adaptation to any dataset regardless of its characteristics.*

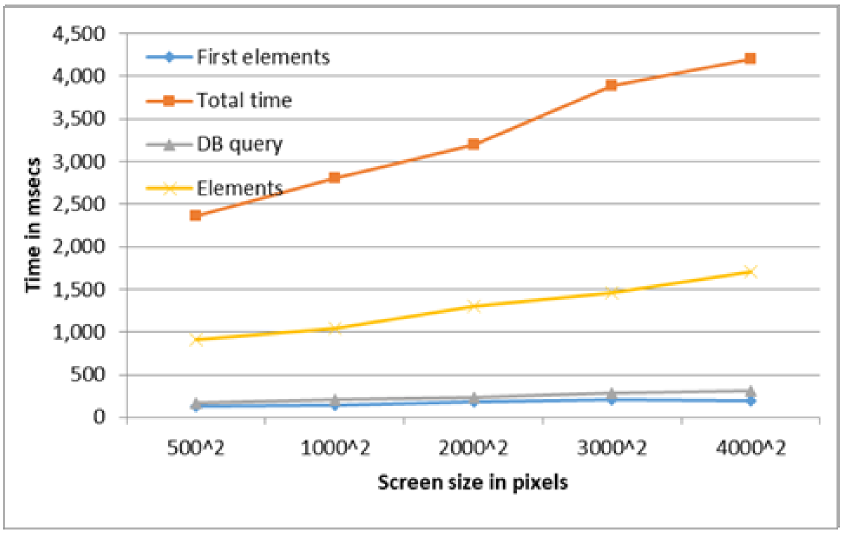
5.3.1 Visualization efficiency

Metrics. *We evaluate the system efficiency based on the visualization response time measured by the time in msecs needed to render the graph elements on the screen after an exploration action. The time presented is the figures is the time needed for the query execution, the rendering time for the first elements to appear on the user screen as the graph is visualized gradually and the total response time of the system when all the elements are rendered. The visualization time is independent of the user actions, zoom, filtering or exploration, but dependent on the number of nodes that are to be rendered.*

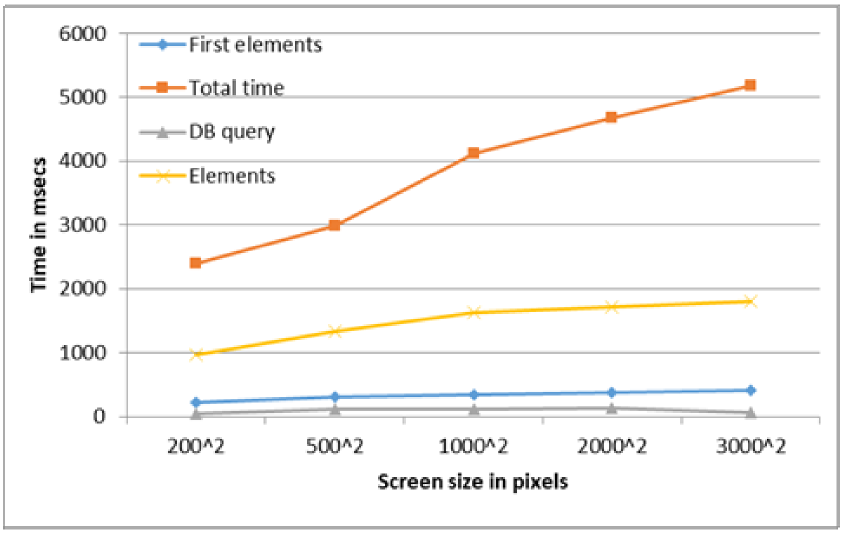
Methodology. *We render randomly selected parts of the datasets using spatial queries with rectangular bounding boxes ranging from* 500*x*500 *px to* 4000*x*4000 *px. As the size of the area increases the spatial queries on the dataset match larger number of graph elements, allowing us to examine the response time over a variation*



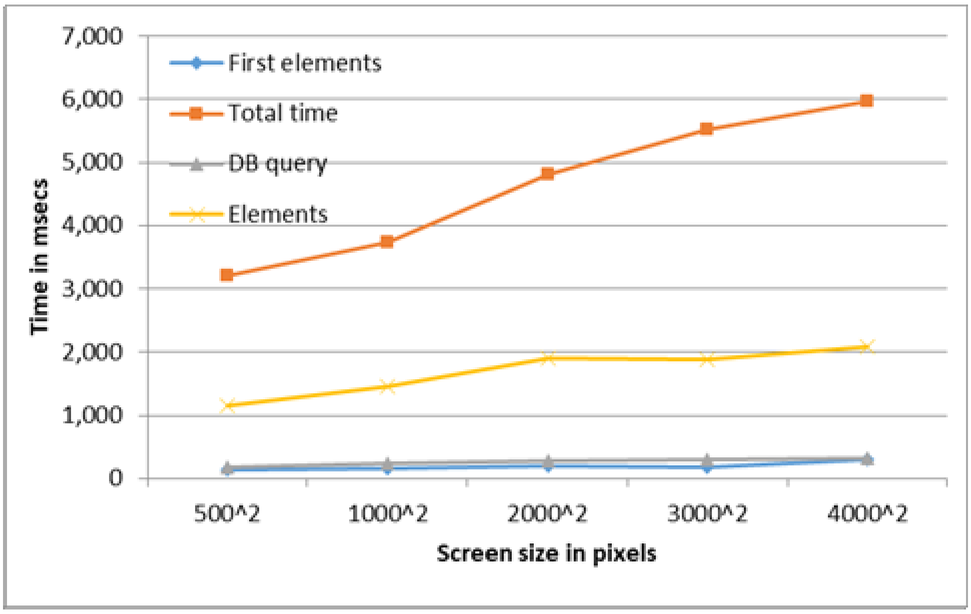
(a) 1M nodes, degree 5



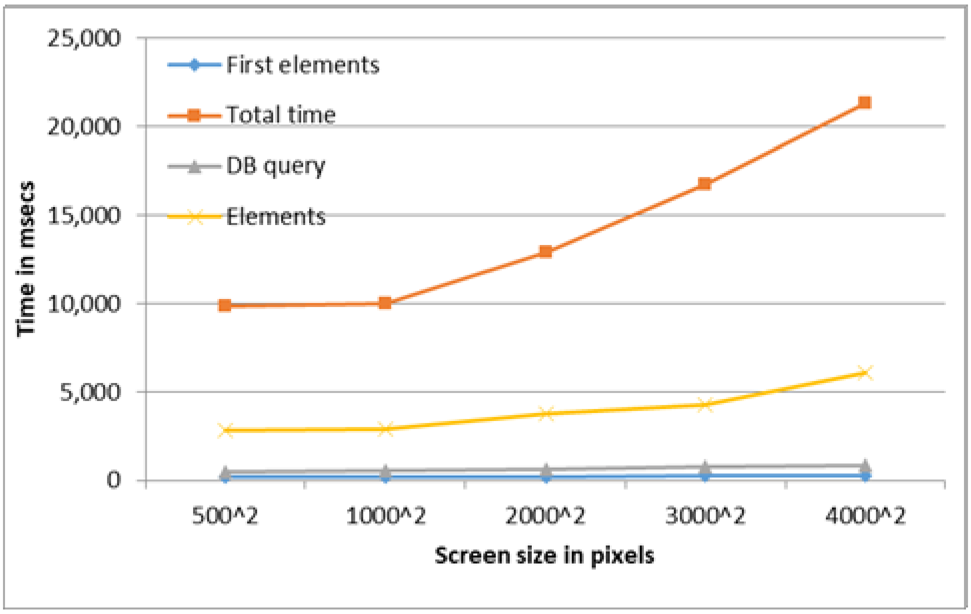
(b) 10M nodes, degree 5



(c) 10M nodes, degree 10

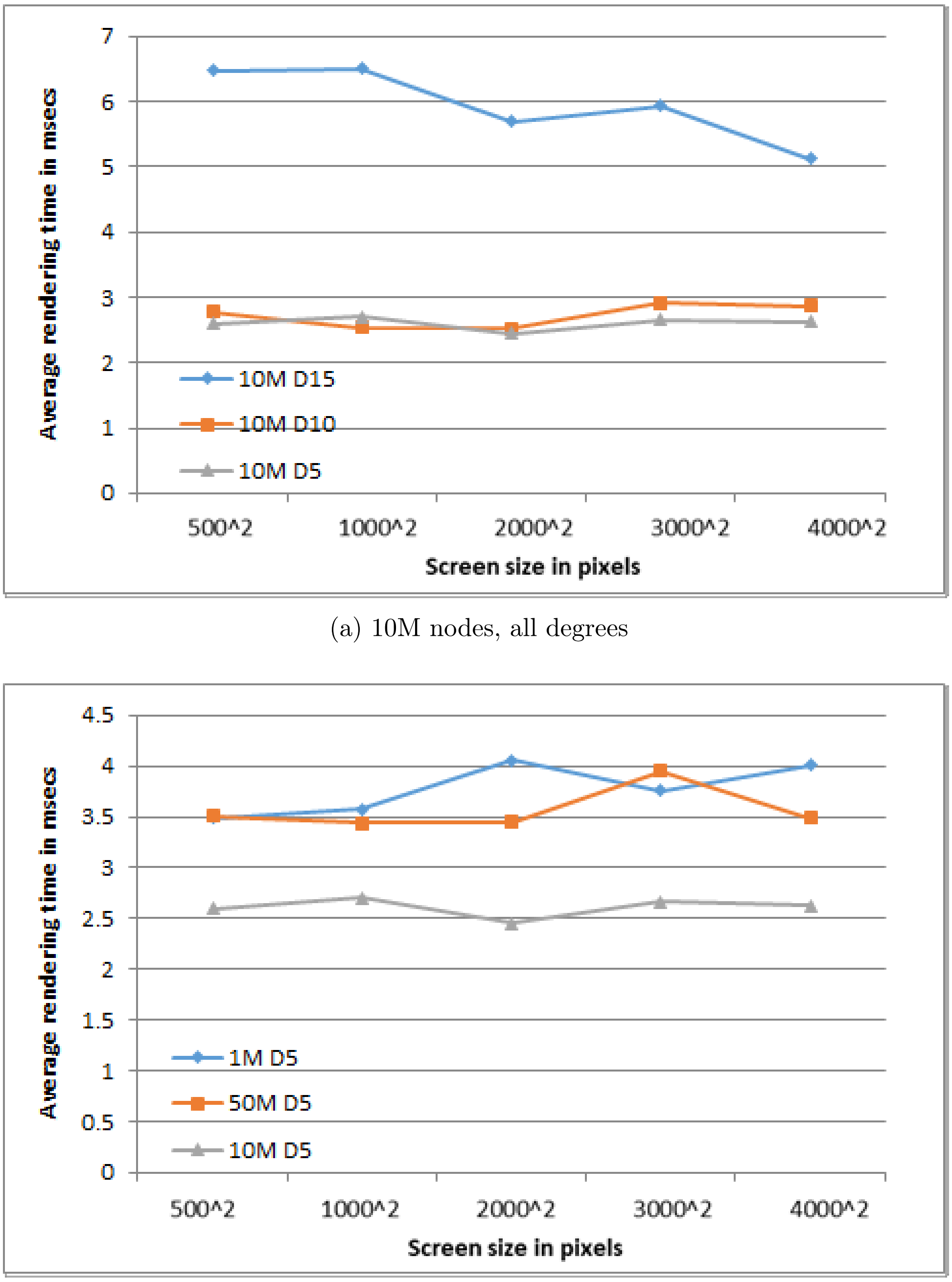


(d) 10M nodes, degree 15



(e) 50M nodes, degree 5

Figure 5.3: Response time for synthetic datasets



(a) 10M nodes, all degrees

(b) degree 5, synthetic datasets

Figure 5.4: Comparison of average response time per graph element

of total rendered graph elements. The experiments present the average results of a series of one hundred repetitions of the graph rendering for each rectangle size.

Results. *In Figure 5.3 we present the results for the synthetic datasets. In all cases the total time is closely connected to the number of rendered elements. The system renders more than* 500 *graph elements in less than two seconds and up to* 5200 *graph elements without causing lagging, performance issues or hindering the user experience, as shown in 5.3 (e). The fact that so many graph elements can be rendered smoothly, is of high importance, as similar systems have limits to the number of presented elements.*

In Figure 5.4 we examine the average time needed for the rendering of one graph element. In Figure 5.4 (a) we show that the average rendering time is not dependent on the density of the input dataset, while in Figure 5.4 (b) we show that it is not dependent on the size of the input dataset.

These experiments prove that our technique scales efficiently for any size or density of the input dataset and supports the exploration of the information for datasets with millions of nodes without any performance issues.

5.3.2 Comparison with other systems

Metrics. *We evaluate the response time of our system by comparing it to that of other visualization systems. Given that not all evaluated systems can visualize the same number of elements, the comparison is based on the average visualization time per graph element.*

Methodology. *First, we examined all the related systems, to identify the most suitable for the comparison. Systems that visualize data in an aggregated manner were not taken into consideration as the aggregation limits the volume of the visualized information and hinder any comparison.*

For the systems FenFire, FlexViz and VOWL 2 [126, 89, 191] the information about accessing the tools was outdated so no experiments could be performed. The systems Tulip and Gephi [17, 19], are available only as desktop application, so they were downloaded and installed for the experimental analysis. The systems RelFinder and FlexViz [130, 45] are available as web applications, so they were accessed directly through the web.

These systems could not handle the volume and the diversity of the datasets used in 5.3.1, so we isolated from the synthetic dataset with 1M nodes and node degree 3, a fully connected graph having 1017 elements. This graph was used to perform experiments with the IVLG system and the two desktop systems that could receive input from a properly formatted file. The web-based systems are not supporting custom dataset so we used for the experiments one of their available datasets.

For the RelFinder tool this was achieved by using the Leipzig/Berlin example that was available and produced a graph of approximately 250 elements. For the LodLive system only a few nodes were rendered as the visualization of each node requires a user interaction. The visualization started from the DBpedia source URI for the word 'paper'.

Results. *As shown in Table 5.4 the IVLG system has the best visualization time per graph element among the systems that was compared with. The Tulip desktop system was the second best, with its performance being dependent on the algorithm selected for the visualization, only the best three results are shown in the table. The*

Table 5.4: Average visualization time per graph element

Tool/Algorithm	**Elements**	**Total time(secs)**	**Element time(msecs)**
IVLG	1017	2.74	2.5
Tulip (Fruchterman Reingold)	1017	4.37	4.29
Tulip (GEM Frick)	1017	4.71	4.63
Tulip (GEM Frick OGDF)	1017	5.28	5.19
Gephi (Yifan Hu)	1017	8.54	8.4
Gephi (Fruchterman Reingold)	1017	17.29	17
Gephi (Force Atlas)	1017	26.43	25.99
RelFinder	247	14.47	58.58
LodLive	1	3.93	3.93(secs)

Gephi system had a lower performance while some of the algorithms could not be calculated due to the available resources of the laptop used.

Again, the three best results from the algorithms successfully tested are presented in the table. RelFinder was tested with the 'skip delayed graph building' option selected; that had a significant impact on the time needed for the graph visualization. LodLive was the slowest system tested as the visualization of each element required almost four seconds.

Conclusions

In this chapter, a novel technique for the pre-processing very large datasets with hundreds of millions of elements and their representation as graphs in the two-dimensional space was presented. The discussed technique has been designed in a way to meet all the identified challenges regarding exploration needs and user experience. The presented technique process large real datasets with millions of elements as well as dense graphs with high node degree. The technique does not impose any restrictions on the dataset while the information is offered through a dedicated API that supports many functionalities, including keyword search, path exploration and neighbor information.

Chapter 6

Targeted Semantic exploration

As more and more datasets become available their utilization in different applications increases in popularity. Their volume and production rate, however, means that their quality and content control is in most cases non-existing, resulting in many datasets that contain inaccurate information of low quality. Especially in the field of conversational assistants, where the datasets come from many heterogeneous sources with no quality assurance, the problem is aggravated. We present here an integrated platform that creates task- and topic-specific conversational datasets to be used for training conversational agents. The platform explores available conversational datasets, extracts information based on semantic similarity and relatedness, and applies a weight-based score function to rank the information based on its value for the specific task and topic. The finalized dataset can then be used for the training of an automated conversational assistance over accurate data of high quality.

Aiming to support the development of conversational automated agents that can carry out accurate, productive and meaningful dialogues we have developed an integrated platform for the creation of semantically correct training datasets. The integrated platform receives as input keywords regarding the topic of interest and explores conversational datasets, extracting semantically related and semantically similar information. The information is then merged, using an innovative weight-based score function, and provided to the user as a topic-specific, complete and strictly structured dataset. The integrated platform has been carefully designed to offer:

- ***Topic-specific information.*** *The platform receives as input a list of predefined categories that represent the main topics that the conversational agent will be asked to handle. If needed, additional keywords can be provided to further customize the semantic topic. This information composes a set of words that are used to determine the semantic similarity and relatedness of the information available to the datasets.*

- ***Task-specific language.*** *The main task that the agent is going to carry out is also provided as input to the platform. This information is used in the weight-based score function in two key ways. On one hand, it is used to determine the balance between the similarity and the relatedness. Agents designed for very specific tasks, and used in ways that limit the conversational topics that they encounter, need datasets that have more similar information due to the limited topic and vocabulary deviations they have to support. Agents that address wider audiences, on the other hand, need to have a larger pool of* *related content to properly respond to questions not directly associated with their main topic. In addition, the task is closely related with the language and vocabulary that should be included in the dataset and used by the score function to support information coming from task-similar datasets. As an example, an agent utilized for answering company's emails is expected to use formal language and proper grammar, so datasets containing social media posts are less relevant to datasets coming from technical support forums.*

- ***Balanced & unbiased dataset.*** *The score function has been carefully designed to produce a ranking that will ensure diversity of information in the output dataset, offering a balanced and unbiased dataset.*
- ***Complete & properly formatted dataset.*** *Information in the input datasets that it is not complete is automatically discarded. The output dataset is always presented to the user as triplets of information connecting a question with the corresponding answer and the semantic score as calculated by the function.*
- ***Ranked output.*** *The processed information is ranked based on the results of the score function. While information that is incomplete, badly formatted and irrelevant, with a score below 50%, is automatically discarded, the rest of the information is available to be used for the training of the conversational agent.*

6.1 System Architecture

We present the system architecture in Figure 6.1. The architecture adheres to a strict data flow that ensures that all input datasets are processed with respect to the parameters defined by the user of the platform. Each question-answer pair is independently given a similarity and a relatedness score. These two values are then merged through a weight-based score function, that takes into consideration the source of the dataset that this pair belongs to as well as the formality of the language that the agent is expected to use and provides a similarity score as a percentage. Based on this score, the available information is ranked, any pair with score below 50% is discarded and the rest are included in the output dataset. The platform is developed as a Java Application.

6.1.1 Input parameters

There are three main input parameters that are required by the integrated platform. The first parameter is a set of words, keywords, that are the main topic of interest for the automated conversational agent. The keywords are a combination of words important for the semantic category chosen by the user and any additional keywords that the user may have provided. The set may contain up to five words, a restriction serving two purposes. On one hand, this ensures that the keywords provided will be carefully selected to reflect only the important semantic meanings that are needed, while at the same time will facilitate the semantic analysis process by ensuring that no false semantic relationships will be identified.

The second parameter that the integrated platform requires as input is the task that the conversational agent will be trained to carry out. This is needed to parameterize the weights used by the score function to ensure that dialogues coming

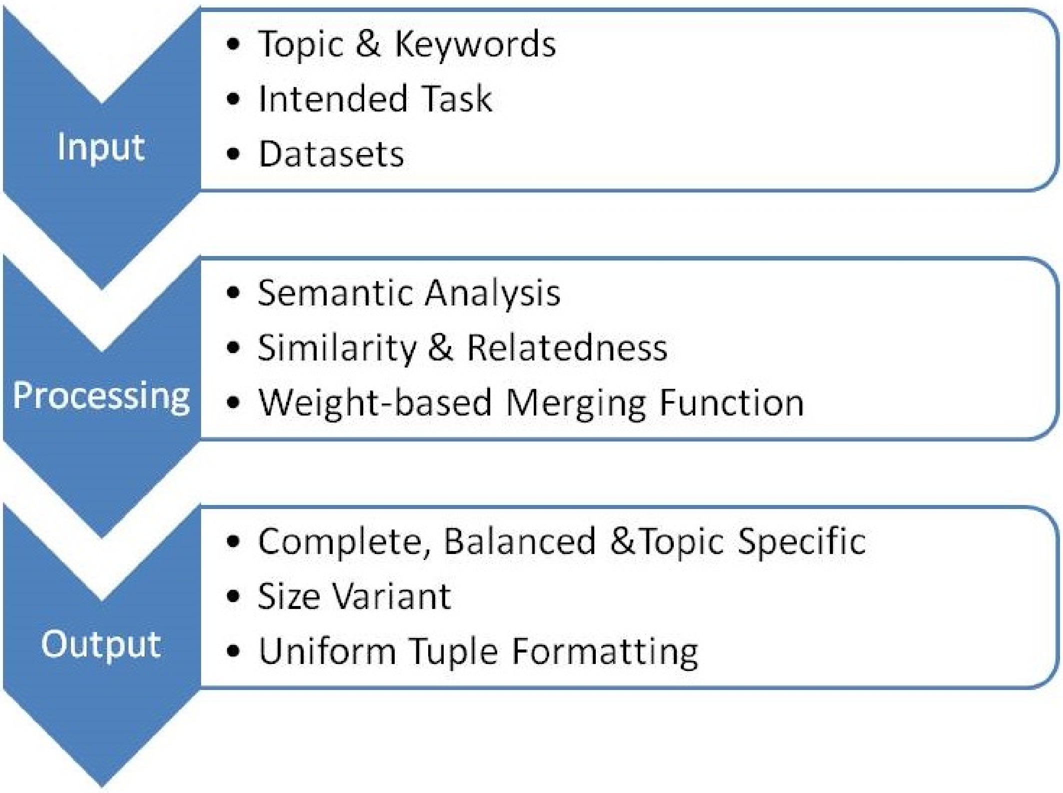

Input
• Topic & Keywords
• Intended Task
• Datasets
Processing
• Semantic Analysis
• Similarity & Relatedness
• Weight-based Merging Function
Output
• Complete, Balanced &Topic Specific
• Size Variant
• Uniform Tuple Formatting

Figure 6.1: System Architecture

from relevant sources will be prioritized. The input is provided in two ways, first the formality of the language is specified, allowing the user to chose between everyday, semi-formal and official language. Additionally, special categories of conversational agents such as social media chatbots or customer support agents are chosen. These categories are one-to-one mapping with the categories that the conversational datasets are coming from, allowing once more the adaptation of the score function in a way that will promote dialogues coming from sources closer to the intended purpose.

The third parameter is the set of datasets that will be explored in order to extract the needed data. Given the plethora of available datasets, it is expected that they offer a high diversity with regard to the quality, the source, the topic and the language, making them the perfect fit for some applications and rendering them unsuitable for others. In addition, users may choose as input a custom dataset provided that it respects the question-answer format.

6.1.2 Main processing

For the purpose of the semantic analysis the WordNet [315] lexical database is used. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms, that are called synsets, each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations, resulting in a network of meaningfully related words and concepts. The dataset offers all the semantic relationships described above.

Based on the above, an overview of the processing data flow is presented here.

Each question-answer pair of the selected datasets is process independently. First a semantic similarity score is calculated, the algorithm is taking into consideration the 'is a' semantic relationships between the input keywords and the examined data to extract a normalized value from 0 *to* 1 *as presented in Section 6.2.1. Next, the semantic relatedness score is calculated utilizing the relationships between sets of words that indicate semantic relatedness to extract again a normalized value from* 0 *to* 1 *as presented in Section 6.2.2. Finally, the weight-based score function merges the two scores, taking into account information about the source dataset and the target task as presented in Section 6.2.3. The result is given as a percentage score.*

6.1.3 Output Dataset

After the processing of the information the question-answer pairs are ranked with regards to their scores. To facilitate further processing and reduce the size of the output, any pairs with score below 0.5 *are discarded as irrelevant to the task and topic. The data are then provided as a Comma-Separated Values file, that contains triplets of information. The triplets include in addition to the question-answer pair the calculated score to facilitate further processing and have the format* $(question, answer, score)$.

6.2 Semantic Analysis

6.2.1 Semantic Similarity

We present here Algorithm 1 that is used to calculate the semantic similarity. The algorithm begins with processing the question-answer pair, extracting the nouns, adjectives, adverbs and verbs, as these are the parts of speech available at the WordNet dataset, creating a set of words that contains the semantic information (Line 3). Next, for each keyword provided with the input a similarity score is calculated (Lines 4-16). Initially (Lines 6-7), the presence of the keyword in the set of words is checked. If the keyword is present the similarity score is set as 1 *and the next keyword is examined. If the keyword is not part of the set of words, it is examined if they share a synset of the WordNet dataset (Lines 9-10). If it is, the similarity score is once again set as* 1*. If it is not, it is examined if there is a generalization between them (Lines 12-16). Hyponymy is given a higher score (Line 16) than hypernymy (Line 13) as it is expected the keywords provided by the user to include general terms associated with the general topic of interest, making phrases that include specifications of these semantically similar. Finally, the score is normalized based on the number of the examined keywords (Line 17).*

6.2.2 Semantic Relatedness

We present here Algorithm 2 that is used to calculate the semantic relatedness. As with the semantic similarity, the algorithm begins with processing the question-answer pair, extracting the nouns, adjectives, adverbs and verbs, creating a set of words that offer semantic information (Line 3). Next, for each keyword provided with the input we calculate a relatedness score (Lines 4-19). Initially (Lines 6-7), it is examined if the keyword is an antonym of the set of words. If it is, then the

Algorithm 1: Similarity score algorithm

```
   Input: $QA$:Question-Answer pair;
          $keywordList$: The list of the keywords provided by the user;
   Output: $similarityScore$: The similarity score between the Question-Answer
           pair and the keyword list;
1  $totalSimilarityScore \leftarrow 0$;                // The sum of the calculated scores
2  $numberOfKeywords \leftarrow 0$;            // The number of the processed keywords
3  $setOfWords \leftarrow extractWords(QA)$;      // Set with nouns, adjectives and
     verbs in the QA
4  foreach $keyword \in keywordList$ do // Find the similarity score for each
     keyword
5      $numberOfKeywords \leftarrow numberOfKeywords + 1$;
6      if $keyword \in setOfWords$ then     // Check if the keyword is part of
         the QA
7          $totalSimilarityScore \leftarrow totalSimilarityScore + 1$ ;
8      else
9          if $IsSynonymy(keyword, setOfWords)$ then
10             $totalSimilarityScore \leftarrow totalSimilarityScore + 1$ ;
11         else
12             if $IsHypernymy(keyword, setOfWords)$ then
13                 $totalSimilarityScore \leftarrow totalSimilarityScore + 0.7$ ;
14             else
15                 if $IsHyponymy(keyword, setOfWords)$ then
16                     $totalSimilarityScore \leftarrow totalSimilarityScore + 0.9$ ;
17 return $similarityScore \leftarrow totalSimilarityScore \div numberOfKeywords$;
```

relatedness score is set as 1*, otherwise it is examined if it is a meronymy (Lines 9-10) and the relatedness score is set as* 0.9*. The hypernymy is again examined (Lines 12-13) and given a relatedness score of* 0.8*. This because, if the keywords is a specific term it is assumed that it was included to emphasize the semantic interest. As a result, question-answer pairs that contain generalizations as expected to have context semantically related as well as similar to the topic of interest. Next, the path between the synsets is calculated (Line 15) and the relatedness score is set as inversely proportional to its length (Line 16). Finally, the score is normalized based on the number of the examined keywords (Line 17).*

6.2.3 Weight-based score function

The weight-based score function is building on the fact that not all data sources are equally important for a specific task. For conversational agents that need to use specific language or perform a specific task it is important to promote question-answer pairs that are coming from relevant sources. As an example, two pairs scoring equally in semantic similarity and relatedness, one coming from a social media dataset and the other from a customer support dataset, do not offer the same value to a conversational agent trained as a chatbot. This is happening as the semantic analysis investigates relationships between two sets of words but does not take into consideration grammar, syntax and formality. Practically, a phrase coming from a social

Algorithm 2: Relatedness score algorithm

```
Input: $QA$:Question-Answer pair;
       $keywordList$: The list of the keywords provided by the user;
Output: $relatednessScore$: The relatedness score between the Question-Answer
        pair and the keyword list;
$totalRelatednessScore \leftarrow 0$;          // The sum of the calculated scores
$numberOfKeywords \leftarrow 0$;          // The number of the processed keywords
$setOfWords \leftarrow extractWords(QA)$;          // Set with nouns, adjectives and
  verbs in the QA
foreach $keyword \in keywordList$ do          // Find the relatedness score for
  each keyword
    $numberOfKeywords \leftarrow numberOfKeywords + 1$;
    if $IsAntonymy(keyword, setOfWords)$ then
        $totalRelatednessScore \leftarrow totalRelatednessScore + 1$ ;
    else
        if $IsMeronymy(keyword, setOfWords)$ then
            $totalRelatednessScore \leftarrow totalRelatednessScore + 0.9$ ;
        else
            if $IsHypernymy(keyword, setOfWords)$ then
                $totalRelatednessScore \leftarrow totalRelatednessScore + 0.8$ ;
            else
                $pathLength \leftarrow countSynsetPath(keyword, setOfWords)$;
                $totalRelatednessScore \leftarrow$
                  $totalRelatednessScore + (1 \div pathLength)$ ;
return $RelatednessScore \leftarrow totalRelatednessScore \div numberOfKeywords$;
```

Table 6.1: Weights for the score function based on language formality and intended task

Task	Similarity	Relatedness	Social Media Dataset	Subtitles/Movies Dataset	Wikipedia Dataset	Yahoo Answers Dataset	Customer Support Datasets	User Provided Datasets
Everyday	0.6	0.4	1	1	0.7	0.8	0.8	1
Semi-formal	0.8	0.2	0.8	0.8	0.9	1	1	1
Formal	0.9	0.1	0.7	0.7	1	1	0.9	1
Chatbot	0.7	0.3	1	0.9	0.8	0.8	0.7	1
Customer Support	0.8	0.2	0.7	0.7	0.8	0.9	1	1
Email correspondence	0.9	0.1	0.6	0.7	1	1	1	1
No preferences specified	0.8	0.2	1	1	1	1	1	1

media dataset will sound closer to what a chatbox is expected to provide than one coming from more a customer support dataset. In addition, it is also very important to have a proper ratio of the similar and related contributions in the output. Conversational agents that have very specific tasks and are expected to encounter a limited and invariable vocabulary need to be trained with question-answer pairs that are very similar to the topic of interest. Contrary, conversational agents that are exposed to uncontrolled environments where they may encounter discussions not strictly related to the topic of interest need to be trained with more broad dataset, one that will include a high percentage of related messages. Based on the above, a weight table has been created that provides the values that should be used in each case as presented in Table 6.1.

These weights are used for two purposes, the dataset related weight calculation which is defining how important the source of information is with regards the task of the conversational agent and the similarity related weight calculation which is

Algorithm 3: Dataset related weight calculation

```
Input: $datasetCategory$: The dataset category;
       $taskSpecification$: The intended task;
       $languageSpecification$: The language formality;
       $weightArray$: The array with all the pre-defined weights;
Output: $totalDatasetWeight$: The calculated weight;
if $taskSpecification$ then
    $taskWeight \leftarrow weightArray[taskSpecification, datasetCategory]$;
    if $languageSpecification$ then    // Calculate dataset specific weight
        $languageWeight \leftarrow weightArray[languageSpecification, datasetCategory]$;
        $totalDatasetWeight \leftarrow \frac{taskWeight+languageWeight}{2}$;
    else
        $totalDatasetWeight \leftarrow taskWeight$;
else
    if $languageSpecification$ then
        $languageWeight \leftarrow weightArray[languageSpecification, datasetCategory]$;
        $totalDatasetWeight \leftarrow languageWeight$;
    else
        $totalDatasetWeight \leftarrow 1$;
return $totalDatasetWeight$
```

Algorithm 4: Similarity related weight calculation

```
Input: $taskSpecification$: The intended task;
       $languageSpecification$: The language formality;
       $weightArray$: The array with all the pre-defined weights;
Output: $totalSimilarityWeight$: The calculated weight;
if $taskSpecification$ then    // Calculate similarity weight
    $taskWeight \leftarrow weightArray[taskSpecification, similarity]$;
    if $languageSpecification$ then
        $languageWeight \leftarrow weightArray[languageSpecification, similarity]$;
        $totalSimilarityWeight \leftarrow \frac{taskWeight+languageWeight}{2}$;
    else
        $totalSimilarityWeight \leftarrow taskWeight$;
else
    if $languageSpecification$ then
        $languageWeight \leftarrow weightArray[languageSpecification, similarity]$;
        $totalSimilarityWeight \leftarrow languageWeight$;
    else
        $totalSimilarityWeight \leftarrow 0.8$;
return $totalSimilarityWeight$
```

Algorithm 5: Weight-based score function

```
Input: $datasetSet$: The input datasets;
       $taskSpecification$: The intended task;
       $languageSpecification$: The language formality;
       $keywordList$: The list of the keywords provided by the user;
       $weightArray$: The array with all the pre-defined weights;
Output: $outputDataset$: A CSV file with triplets of information;
$totalSimilarityWeight \leftarrow similarityWeightCalculation$
  $(taskSpecification, languageSpecification, weightArray)$;
$totalRelatednessWeight \leftarrow (1 - totalSimilarityWeight)$;
foreach $dataset \in datasetSet$ do
    $totalDatasetWeight \leftarrow datasetWeightCalculation$
      $(datasetCategory, taskSpecification,$
      $languageSpecification, weightArray)$;
    foreach $QApair \in dataset$ do        // Find the total score for each QA
        $similarityScore \leftarrow calculateSimilarity(QApair, keywordList)$;
        $similarityScore \leftarrow similarityScore \times totalSimilarityWeight$;
        $relatednessScore \leftarrow calculateRelatedness(QApair, keywordList)$;
        $relatednessScore \leftarrow relatednessScore \times totalRelatednessWeight$ ;
        $totalScore \leftarrow$
          $(similarityScore + relatednessScore) \times totalDatasetWeight$;
        if $totalScore \geq 0.5$ then        // Check if the total score is above
          threshold
            $outputDataset \leftarrow outputDataset + (QApair, totalScore)$;
return $outputDataset$
```

defining the ratio of the related messages that will be included int he output dataset.

As presented in Algorithm 3, in order to calculate the weight for a specific dataset the choices of the user are examined. If a user has defined a task (Lines 1-7) then the task weight is retrieved from the weight table (Line 2). If the user has also defined a language formality level (Lines 3-5) then the average of the two weights is set as the overall dataset weight (Line 5). If the user has provided only a language weight then this is the final weight for the dataset (Lines 9-11). Finally, in the case that there are no user preferences the dataset level is set as 1 *(Line 13).*

The similarity related weight, as presented in Algorithm 4, is calculated again by examining the choices of the users. If a user has defined a task (Lines 1-7) then the task weight is retrieved from the weight table (Line 2). If the user has also defined a language formality level (Lines 3-5) then the average of the two weights is set as the overall similarity weight (Line 5), if not then the similarity is defined only by the selected task (Line 7). If the user has provided only a language weight then this is the final similarity weight (Lines 9-11). Finally, in the case that there are no user preferences the similarity is set as 0.8 *(Line 13), ensuring that enough related information will be part of the dataset without disturbing the semantic focus.*

Bringing all the above together is the weight-based score function, presented in Algorithm 5. The function receives as input the user preferences, a list of datasets, a list of keywords and the array with the pre-defined weights and produces a CSV file with triplets of information, the question-answer pair and the calculated score. First, the similarity weight (Line 1) is calculated based on the user preferences and the weight array. Next, the relatedness weight (Line 2) is calculated, based on the similarity weight. For each dataset (Line 3-12), the dataset specific weight is calculated (Line 4). Then for each question-answer pair in the dataset, a similarity (Line 6) and a relatedness (Line 8) score is calculated, the scores are adjusted based on the calculated weights (Line 7 & 9). The two scores are added together and multiplied by the dataset weight (Line 10). Finally, if the final score for the pair is above 0.5 *it is added to the output file along with the calculated score (Line 11-12).*

Example. *Given the requirements for a formal, customer support agent with the additional keyword of device, we examine the question-answer pair, coming from a customer support dataset:*
Q: Is your phone working now?
A: Yes, restarting it solved the problem.
Starting with Algorithm 5, the $totalSimilarityWeight$ *is calculated as* $(0.9+0.8)/2$*, and the* $totalRelatednessWeight$ *is set as* 0.15*. Then the* $totalDatasetWeight$ *is calculated as* $(0.9+1)/2$*. This pair produces a set of words {phone, working, restarting, solved, problem} while the keywords are {device, work, solve, problem}. For these sets Algorithm 4 provides a similarity score of* $(0.7+1+1+1)/4$ *and Algorithm 2 provides a relatedness score of* $(0.8+0+0+0)/4$*, so the total score for this pair, taking into consideration the calculated weights, is* $(0.93*0.85+0.2*0.15)*0.95 = 0.78$*. This pair is semantically relevant to the topic and task of the agent so it will be part of the output dataset.*

6.3 Experimental Analysis

In order to evaluate the proposed algorithms and the potential improvement in the training of conversational assistants we have conducted a series of experimental anal-

ysis. The experiments were conducted using a laptop with an Intel(R) Core(TM) i7-4500U CPU at 1.80GHz, 4GB RAM memory connected to a 12Mbps home network connection. Python version 3.6 was used as well as the nltk and scikit-learn libraries along with their dependencies were used in order to implement the algorithms discussed in Section 13.

Datasets. *The latest dump of the Wikipedia was used as the base dataset for all the experiments. The dump enwiki-latest-pages-articles.xml.bz2, containing the latest pages of all the articles of Wikipedia, was downloaded from* `https://dumps.wikimedia.org/enwiki/latest/`*. The dump available in March 2020 was 16.02 GB and it contained* 10682757 *pages, from these some were indexing or disambiguation pages that were excluded from the dataset. The final dataset contained* 6062172 *article pages. These pages were further processed in order to eliminate special characters, templates, tables and and other elements available in the dump related with the formatting, the images, the references and the annotations available in the pages.*

Next, articles that had less than two hundred characters were removed from the dataset, as these articles were mainly placeholders for topics that needed to be further populated and/or articles that were only contained redirection to other articles, offering no additional information. The final dataset which included only the plain text of the pages was merged into one file with size of $13.3GB$.

All the experiments presented below are using as a basis this dataset. This choice serves two main purposes. On the one hand, by using only one dataset we are eliminating the subjectivity of the dataset from the experiments contacted. All the metrics can be compared between them without the interference of the difference between the used dataset. In addition, any bias introduced to the dataset during the pre-processing by the elimination of certain articles and content will be uniform for all the contacted experiments and will not affect the comparison of the results. On the other hand, the volume and topic diversity of the Wikipedia dataset allows us to create many topic specific datasets and showcase the adaptivity of our technique. At the same time it allows us to emphasize the importance of limiting the input dataset based on the application needs in order to improve the performance and the user experience.

For the needs of the experiments below we chose to use the complete dataset as the basis for comparison and create four new ones on different topics. The topics and keywords chosen are:

- *Tennis(1), with keywords {'tennis', 'ball', 'court', 'sport', 'racket'}*
- *Tennis(2), with keywords {'tennis', 'ball', 'sport', 'racket'}*
- *Animals, with keywords {'animal', 'dog', 'jungle', 'cat', 'lion'}*
- *Food, with keywords {'food', 'meat', 'vegetable', 'fruit', 'fish'}*

The topics for the datasets were selected based on two criteria. The topics had to be at the same time specific enough with regard to the overall Wikipedia content and broad enough to support a possible application and allow the testing of multiple related questions.

Keyword selection & Ambiguation. *As discussed in Section 6.1.1 the keywords selected are very important and will determine if the produced dataset can be* *used for the intended purpose and what would be its performance. Choosing keywords that are too specific will result in less words being connected with the with the examined semantic relationships, making hard for phrases to have a high score unless they are also very specific to the keyword, resulting in a smaller dataset, more focused on the topic and the semantic neighbour of the keywords. Choosing keywords that are too broad will create a large pool of words semantically associated with them, resulting in a larger dataset that will cover a wider semantic area.*

While we encourage the use of the algorithms with balanced keywords, that represent the main semantic categories of the topic, both options for specific and broad keywords may be preferable based on the task that the dataset will be used for. As an example, a technical support chatbot for mobile phones needs to have very specific keywords and a limited dataset while a personal assistant chatbox needs to have a larger dataset with broader keywords.

In all cases, extra care should be given to words that have multiple meanings or their main usage is not the one related to the semantic topic of interest. Such an example is shown here, with the creation of two datasets for the semantic topic of Tennis. The first set of keywords contains the word court which is a reference to the tennis court, the playing field for tennis. This keyword, however, is primarily used to identify the court of law and it is tightly associated with the judicial system. This is evident from the synonyms of the word court as they are identified by the WordNet lexical database, which are:
['courtyard', 'motor_inn', 'Court', 'court_of_justice', 'homage', '', 'Margaret_Court', 'court_of_law', 'motor_lodge', 'motor_hotel', 'tribunal', 'law_court', 'tourist_court', 'royal_court', 'court', 'courtroom'].

It is also worth noticing that if the term is referred as 'tennis court' which is the most probable scenario, the addition of the keyword court as a standalone at the list of keywords is not expected to enrich the dataset with further tennis related material that the keyword tennis on its own would fail to include.

The role of this term to the dataset size and performance will be evaluated at the following Sections and discussed accordingly.

6.3.1 Weights vs Output dataset size

Metrics. *The first step for the evaluation of our technique is to measure how the weights discussed in Section 6.2.3 affect the volume of the produced dataset. This is very crucial as the volume of the dataset will not only affect the quality of the responses of a chatbox using it but also the time needed to provide the proper response.*

Methodology. *For the four topics, and the identified keywords, we run the algorithms for multiple weights between similarity and relatedness as well as different thresholds. The combinations are shown in Table 6.2. For now on, aiming to present the results in a more compact way this information will be presented as (similarity/relatedness/threshold), meaning that (0.7/0.3/0.8) will characterize a dataset created with similarity weight of 0.7, relatedness weight on 0.3 and a threshold of 0.8.*

Results. *In Table 6.3 we present the sizes of the datasets created for the four topics in the examined combinations of weights and thresholds. While the sizes of the datasets for the tennis topic is remarkably smaller than the ones for the more*

Table 6.2: Similarity & Relatedness examined weight combinations and thresholds

Similarity	**Relatedness**	**Threshold**
0.7	0.3	0.7
0.7	0.3	0.8
0.7	0.3	0.9
0.7	0.3	0.95
0.8	0.2	0.7
0.8	0.2	0.8
0.8	0.2	0.9
0.8	0.2	0.95
0.9	0.1	0.7
0.9	0.1	0.8
0.9	0.1	0.9
0.9	0.1	0.95

Table 6.3: Output dataset sizes based on weights & thresholds

Characteristics	**Tennis(1)**	**Tennis(2)**	**Animals**	**Food**
(0.7/0.3/0.7)	2.5 MB	2.1 MB	11.3 MB	15.9 MB
(0.7/0.3/0.8)	2.3 MB	1.9 MB	10.6 MB	14.8 MB
(0.7/0.3/0.9)	676.5 kB	594.2 kB	1.8 MB	2.1 MB
(0.7/0.3/0.95)	590 kB	432.1 kB	1.6 MB	1.8 MB
(0.8/0.2/0.7)	95.6 MB	91.2 MB	538.2 MB	649.4 MB
(0.8/0.2/0.8)	2.5 MB	2.1 MB	11.2 MB	15.7 MB
(0.8/0.2/0.9)	2.3 MB	1.9 MB	10.5 MB	14.5 MB
(0.8/0.2/0.95)	590 kB	432.1 kB	1.6 MB	1.8MB
(0.9/0.1/0.7)	115.4 MB	112.1 MB	612.1 MB	701 MB
(0.9/0.1/0.8)	111.2 MB	97.5 MB	583.4 MB	692.9 MB
(0.9/0.1/0.9)	677 kB	612 kB	1.9 MB	2.2 MB
(0.9/0.1/0.95)	590 kB	432.1 kB	1.6 MB	1.8 MB

general categories there are still some interesting comparison among the results.

To begin with, it is remarkable that the (0.7/0.3/0.7) and the (0.8/0.2/0.8) datasets as well as the (0.7/0.3/0.8) and the (0.8/0.2/0.9) datasets are almost the same size. A manual inspection of their content showed that the content was also very similar. This lead to the assumption that the similarity plays a key role to the overall score and that the relatedness only has a small part of the balance.

Next, the fact that the (0.9/0.1/0.9) dataset do not follow this pattern as well as the similar size for the (x/x/0.95) datasets leads to the assumption that it is very difficult to find phrases that score very highly. It is expected that most phrases included in these datasets will contain one of the given keywords. Finally, as it was expected, the Tennis(2) dataset which was produced without the court keyword in the given list, was significant smaller that the Tennis(1) dataset. The affect that this has in the response quality is discussed in the next section.

6.3.2 Response quality evaluation

Metrics. *In our to evaluate the impact our approach has to the response quality we are going to use the cosine similarity. Cosine similarity measures the similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. It takes advantage of the fact that the cosine between to vectors that have 0 degrees of angle will be 1 and as the angle increases the value will decrease as well.*

In the world of semantic similarity the cosine similarity is very important for two key reasons. On the one hand, the metric is based only on the vectors orientation and not magnitude, meaning that the proportional size of the question to the answer will not affect the value of the metric. On the other hand, the cosine similarity has an outcome is neatly bounded in the [0,1] space allowing the easy comparison between questions, datasets and topics.

Methodology. *For each topic we are going to define three questions, one with multiple keywords, one with only one keyword and one without any of the given keywords. In addition for the tennis topic we are going to define an additional question with the keyword court as it would appear in a judicial related question. For these questions we are going to record the cosine similarity of the best response, the average similarity for the top five, ten and twenty responses for the complete dataset of* $13.3GB$ *as well as the datasets produced by our algorithms.*

Results. *We present here, the questions asked to the datasets as well as the obtained results. For the tennis topic the questions that were used, are:*

- *“Which is the most famous tennis tournament?", results are presented in Table 6.4 and in Table 6.8.*
- *“What are the requirements for the balls and rackets used during a tennis match?", results are presented in Table 6.5 and in Table 6.9.*
- *“Which is the most famous player of all times?", results are presented in Table 6.6 and in Table 6.10.*
- *“In the court of law how many who makes the decisions?", results are presented in Table 6.7 and in Table 6.11.*

For the animal topic the questions that were used, are:

- *“Which is the most engaged animal in the world?", results are presented in Table 6.12.*
- *“What are the chances of meeting a lion when walking in a jungle?", results are presented in Table 6.13.*
- *“Which is the most famous zoo of all times?", results are presented in Table 6.14.*

For the food topic the questions that were used, are:

- *“Which is the most popular food in the world?", results are presented in Table 6.15.*

- *“What are the the nutritional needs of a person for meat, fruits, vegetables and fish?", results are presented in Table 6.16.*
- *“Which is the most famous chef of all times?", results are presented in Table 6.17.*

From the results presented in the tables, presenting the average cosine similarity of the datasets for the chosen questions there are many interesting results.

To begin with, it is proven that the similarities between file sizes as discussed in Section 6.3.1 are extended in the quality of the content of the file as well as the recorded cosine similarity. For the topic of tennis, which is relatively targeted the cosine similarity was almost identical between combinations of weights and thresholds that had similar file sizes. For the more general subjects of animals and food, there were some more noticeable differences, showing how the relatedness affected the content of the dataset for for general cases. In this case too , however, the differences could be considered insignificant when compared with differences to other weights and thresholds.

Next, it is important to note that in most cases the (0.9,0.1,0.9) combination seems to provide the best overall results for the average similarity score. This comes as no surprise, as the restrictive size of the datasets does not allows semantically dissimilar phrases to be infiltrate the top suggested responses of the result.

Furthermore, focusing on Tables 6.7 and 6.11 we note that the presence of the keyword court in the formation of one of the datasets for the tennis topic has some significant affect to the performance of this dataset to a judicial related question. Also, in both cases the general dataset, with all the identified articles, has better cosine similarity score. This shows that the one keyword was not enough to to cover the topic of law and that significant amount of phrases containing judicial related content were excluded.

In addition, examining the performance of the tennis related datasets to the other questions, it is remarkable that they have the same or very similar cosine scores. This proves that proposed technique has some resilience and if one of the provided keywords is not well-chosen this will affect the size of the dataset but it won’t derail it from its semantic focus and overall performance.

6.3.3 Dataset size vs Response time

Metrics. *In order to evaluate the impact of our approach to the user experience we are also examining the time needed for the calculation of the cosine similarity as discussed in the previous experiment.*

Methodology. *For each question we are recording the time needed for the creation of the vectors for the input dataset and their comparison with the vector of the question. The time needed is also presented in the tables discussed above along with the cosine similarities.*

Results. *The results show that there is a direct correlation between the size of the dataset and the time needed to obtain the cosine similarity comparison results. While for some cases, especially pre-processing services and offline training, the time needed might not be directly affecting the user experience it does affect the overall performance of the system. It also affects the needs for memory and computational power. This may be counter-productive in large scare set-ups, especially taking into* *consideration that carefully limiting the input dataset by applying the proposed technique will not affect the result quality.*

Conclusions

Recognizing that Big Data often lack the needed quality for the training of artificial intelligence agents, an integrated platform was presented in this chapter that performs semantic analysis in conversational dataset in order to create high quality datasets for the training of conversational agents. The input question-answer pairs are examined based on their semantic similarity, their semantic relatedness and their source to extract a score that allow us to rank them based on their potential contribution for the specific task and topic.

Table 6.4: Cosine Similarities comparison for tennis dataset, with court keyword, question 1

	(0.7/0.3/0.7)	(0.7/0.3/0.8)	(0.7/0.3/0.9)	(0.8/0.2/0.7)	(0.8/0.2/0.8)	(0.8/0.2/0.9)	(0.9/0.1/0.7)	(0.9/0.1/0.8)	(0.9/0.1/0.9)	Complete
1st	0.6613	0.6843	0.6951	0.6423	0.6613	0.6843	0.6352	0.6423	0.6951	0.6754
Top 5	0.4559	0.4784	0.4952	0.4472	0.4559	0.4784	0.4251	0.4472	0.4988	0.4752
Top 10	0.4288	0.4358	0.4467	0.4009	0.4288	0.4358	0.3754	0.4009	0.4604	0.4159
Top 20	0.3803	0.3921	0.4105	0.3629	0.3803	0.3921	0.3265	0.3629	0.4261	0.3998
Time	17.0754	16.549	11.0287	654.2	17.0754	16.549	694.2	759.59	11.2561	28.1h

Table 6.5: Cosine Similarities comparison for tennis dataset, with court keyword, question 2

	(0.7/0.3/0.7)	(0.7/0.3/0.8)	(0.7/0.3/0.9)	(0.8/0.2/0.7)	(0.8/0.2/0.8)	(0.8/0.2/0.9)	(0.9/0.1/0.7)	(0.9/0.1/0.8)	(0.9/0.1/0.9)	Complete
1st	0.5453	0.5587	0.5657	0.5323	0.5453	0.5587	0.5143	0.5323	0.5827	0.5476
Top 5	0.364	0.3897	0.3977	0.351	0.364	0.3897	0.339	0.351	0.4077	0.3854
Top 10	0.3754	0.3857	0.3937	0.3654	0.3754	0.3857	0.3584	0.3654	0.4117	0.3652
Top 20	0.3576	0.3691	0.3841	0.3446	0.3576	0.3691	0.3396	0.3446	0.3961	0.3584
Time	17.0754	16.549	11.0287	654.2	17.0754	16.549	694.2	759.59	11.2561	28.1h

Table 6.6: Cosine Similarities comparison for tennis dataset, with court keyword, question 3

	(0.7/0.3/0.7)	(0.7/0.3/0.8)	(0.7/0.3/0.9)	(0.8/0.2/0.7)	(0.8/0.2/0.8)	(0.8/0.2/0.9)	(0.9/0.1/0.7)	(0.9/0.1/0.8)	(0.9/0.1/0.9)	Complete
1st	0.4239	0.4624	0.4744	0.4049	0.4239	0.4624	0.3939	0.4049	0.4944	0.4752
Top 5	0.2626	0.3157	0.3207	0.2496	0.2626	0.3157	0.2446	0.2496	0.3367	0.3158
Top 10	0.2566	0.2866	0.3066	0.2426	0.2566	0.2866	0.2376	0.2426	0.3216	0.2895
Top 20	0.2433	0.2547	0.2677	0.2253	0.2433	0.2547	0.2083	0.2253	0.2817	0.2742
Time	17.0754	16.549	11.0287	654.2	17.0754	16.549	694.2	759.59	11.2561	28.1h

Table 6.7: Cosine Similarities comparison for tennis dataset, with court keyword, question 4

	(0.7/0.3/0.7)	(0.7/0.3/0.8)	(0.7/0.3/0.9)	(0.8/0.2/0.7)	(0.8/0.2/0.8)	(0.8/0.2/0.9)	(0.9/0.1/0.7)	(0.9/0.1/0.8)	(0.9/0.1/0.9)	Complete
1st	0.3922	0.412	0.417	0.3782	0.3922	0.412	0.3632	0.3782	0.436	0.5965
Top 5	0.2639	0.2986	0.3146	0.2439	0.2639	0.2986	0.2249	0.2439	0.3246	0.4925
Top 10	0.2612	0.2745	0.2915	0.2442	0.2612	0.2745	0.2342	0.2442	0.3075	0.3675
Top 20	0.2467	0.2551	0.2731	0.2287	0.2467	0.2551	0.2137	0.2287	0.2781	0.3162
Time	17.0754	16.549	11.0287	654.2	17.0754	16.549	694.2	759.59	11.2561	28.1h

Table 6.8: Cosine Similarities comparison for tennis dataset, without court keyword, question 1

	(0.7/0.3/0.7)	***(0.7/0.3/0.8)***	***(0.7/0.3/0.9)***	***(0.8/0.2/0.7)***	***(0.8/0.2/0.8)***	***(0.8/0.2/0.9)***	***(0.9/0.1/0.7)***	***(0.9/0.1/0.8)***	***(0.9/0.1/0.9)***	***Complete***
1st	*0.6613*	*0.6843*	*0.6951*	*0.6423*	*0.6613*	*0.6843*	*0.6352*	*0.6423*	*0.6951*	*0.6754*
Top 5	*0.4559*	*0.4784*	*0.4952*	*0.4472*	*0.4559*	*0.4784*	*0.4251*	*0.4472*	*0.4988*	*0.4752*
Top 10	*0.4288*	*0.4358*	*0.4467*	*0.4009*	*0.4288*	*0.4358*	*0.3754*	*0.4009*	*0.4604*	*0.4159*
Top 20	*0.3803*	*0.3921*	*0.4105*	*0.3629*	*0.3803*	*0.3921*	*0.3265*	*0.3629*	*0.4261*	*0.3998*
Time	*16.531*	*15.9525*	*9.5634*	*625.2*	*15.5234*	*15.0631*	*618.7*	*711.65*	*9.5858*	*28.1h*

Table 6.9: Cosine Similarities comparison for tennis dataset, without court keyword, question 2

	(0.7/0.3/0.7)	***(0.7/0.3/0.8)***	***(0.7/0.3/0.9)***	***(0.8/0.2/0.7)***	***(0.8/0.2/0.8)***	***(0.8/0.2/0.9)***	***(0.9/0.1/0.7)***	***(0.9/0.1/0.8)***	***(0.9/0.1/0.9)***	***Complete***
1st	*0.5453*	*0.5587*	*0.5657*	*0.5323*	*0.5453*	*0.5587*	*0.5143*	*0.5323*	*0.5827*	*0.5476*
Top 5	*0.364*	*0.3897*	*0.3977*	*0.351*	*0.364*	*0.3897*	*0.339*	*0.351*	*0.4077*	*0.3854*
Top 10	*0.3754*	*0.3857*	*0.3937*	*0.3654*	*0.3754*	*0.3857*	*0.3584*	*0.3654*	*0.4117*	*0.3652*
Top 20	*0.3576*	*0.3691*	*0.3841*	*0.3446*	*0.3576*	*0.3691*	*0.3396*	*0.3446*	*0.3961*	*0.3584*
Time	*16.531*	*15.9525*	*9.5634*	*625.2*	*15.5234*	*15.0631*	*618.7*	*711.65*	*9.5858*	*28.1h*

Table 6.10: Cosine Similarities comparison for tennis dataset, without court keyword, question 3

	(0.7/0.3/0.7)	***(0.7/0.3/0.8)***	***(0.7/0.3/0.9)***	***(0.8/0.2/0.7)***	***(0.8/0.2/0.8)***	***(0.8/0.2/0.9)***	***(0.9/0.1/0.7)***	***(0.9/0.1/0.8)***	***(0.9/0.1/0.9)***	***Complete***
1st	*0.4239*	*0.4624*	*0.4744*	*0.4049*	*0.4239*	*0.4624*	*0.3939*	*0.4049*	*0.4944*	*0.4752*
Top 5	*0.2626*	*0.3157*	*0.3207*	*0.2496*	*0.2626*	*0.3157*	*0.2446*	*0.2496*	*0.3367*	*0.3158*
Top 10	*0.2566*	*0.2866*	*0.3066*	*0.2426*	*0.2566*	*0.2866*	*0.2376*	*0.2426*	*0.3216*	*0.2895*
Top 20	*0.2433*	*0.2547*	*0.2677*	*0.2253*	*0.2433*	*0.2547*	*0.2083*	*0.2253*	*0.2817*	*0.2742*
Time	*16.531*	*15.9525*	*9.5634*	*625.2*	*15.5234*	*15.0631*	*618.7*	*711.65*	*9.5858*	*28.1h*

Table 6.11: Cosine Similarities comparison for tennis dataset, without court keyword, question 4

	(0.7/0.3/0.7)	***(0.7/0.3/0.8)***	***(0.7/0.3/0.9)***	***(0.8/0.2/0.7)***	***(0.8/0.2/0.8)***	***(0.8/0.2/0.9)***	***(0.9/0.1/0.7)***	***(0.9/0.1/0.8)***	***(0.9/0.1/0.9)***	***Complete***
1st	*0.2722*	*0.298*	*0.3915*	*0.3032*	*0.2772*	*0.297*	*0.3052*	*0.3292*	*0.421*	*0.5965*
Top 5	*0.2469*	*0.2826*	*0.2946*	*0.2279*	*0.2439*	*0.2786*	*0.2109*	*0.2299*	*0.3116*	*0.4925*
Top 10	*0.2312*	*0.2585*	*0.2745*	*0.2192*	*0.2402*	*0.2545*	*0.2042*	*0.2242*	*0.2925*	*0.3675*
Top 20	*0.2367*	*0.2421*	*0.2561*	*0.2137*	*0.2287*	*0.2401*	*0.1947*	*0.2177*	*0.2681*	*0.3162*
Time	*16.531*	*15.9525*	*9.5634*	*625.2*	*15.5234*	*15.0631*	*618.7*	*711.65*	*9.5858*	*28.1h*

Table 6.12: Cosine Similarities comparison for animal dataset, question 1

	(0.7/0.3/0.7)	***(0.7/0.3/0.8)***	***(0.7/0.3/0.9)***	***(0.8/0.2/0.7)***	***(0.8/0.2/0.8)***	***(0.8/0.2/0.9)***	***(0.9/0.1/0.7)***	***(0.9/0.1/0.8)***	***(0.9/0.1/0.9)***	***Complete***
1st	*0.6672*	*0.687*	*0.687*	*0.6532*	*0.6652*	*0.685*	*0.6342*	*0.6492*	*0.706*	*0.685*
Top 5	*0.5369*	*0.5736*	*0.5846*	*0.5189*	*0.5389*	*0.5686*	*0.4969*	*0.5149*	*0.5946*	*0.5775*
Top 10	*0.5362*	*0.5445*	*0.5625*	*0.5172*	*0.5362*	*0.5475*	*0.5052*	*0.5162*	*0.5775*	*0.5695*
Top 20	*0.5167*	*0.5281*	*0.5461*	*0.5037*	*0.5177*	*0.5291*	*0.4857*	*0.5027*	*0.5521*	*0.5392*
Time	*152.6*	*148.9*	*18.9*	*17.6*	*3528*	*185.4*	*128.3*	*4028.5*	*25.6*	*28.1h*

Table 6.13: Cosine Similarities comparison for animal dataset, question 2

	(0.7/0.3/0.7)	***(0.7/0.3/0.8)***	***(0.7/0.3/0.9)***	***(0.8/0.2/0.7)***	***(0.8/0.2/0.8)***	***(0.8/0.2/0.9)***	***(0.9/0.1/0.7)***	***(0.9/0.1/0.8)***	***(0.9/0.1/0.9)***	***Complete***
1st	*0.6552*	*0.678*	*0.685*	*0.6402*	*0.6542*	*0.682*	*0.6262*	*0.6382*	*0.704*	*0.684*
Top 5	*0.5259*	*0.5666*	*0.5826*	*0.5069*	*0.5239*	*0.5636*	*0.4949*	*0.5089*	*0.5886*	*0.5625*
Top 10	*0.5252*	*0.5435*	*0.5595*	*0.5072*	*0.5212*	*0.5445*	*0.5002*	*0.5132*	*0.5745*	*0.5585*
Top 20	*0.5147*	*0.5181*	*0.5411*	*0.4967*	*0.5077*	*0.5201*	*0.4757*	*0.4967*	*0.5461*	*0.5342*
Time	*152.6*	*148.9*	*18.9*	*17.6*	*3528*	*185.4*	*128.3*	*4028.5*	*25.6*	*28.1h*

Table 6.14: Cosine Similarities comparison for animal dataset, question 3

	(0.7/0.3/0.7)	***(0.7/0.3/0.8)***	***(0.7/0.3/0.9)***	***(0.8/0.2/0.7)***	***(0.8/0.2/0.8)***	***(0.8/0.2/0.9)***	***(0.9/0.1/0.7)***	***(0.9/0.1/0.8)***	***(0.9/0.1/0.9)***	***Complete***
1st	*0.5582*	*0.5766*	*0.5798*	*0.5392*	*0.5532*	*0.5778*	*0.5262*	*0.5432*	*0.5997*	*0.6768*
Top 5	*0.4292*	*0.4636*	*0.4756*	*0.4079*	*0.4269*	*0.4616*	*0.3869*	*0.4049*	*0.4896*	*0.5647*
Top 10	*0.4232*	*0.4379*	*0.4565*	*0.4072*	*0.42245*	*0.4385*	*0.3952*	*0.4075*	*0.4706*	*0.5605*
Top 20	*0.4125*	*0.4167*	*0.4348*	*0.3927*	*0.4119*	*0.4215*	*0.3787*	*0.3937*	*0.4391*	*0.5302*
Time	*152.6*	*148.9*	*18.9*	*17.6*	*3528*	*185.4*	*128.3*	*4028.5*	*25.6*	*28.1h*

Table 6.15: Cosine Similarities comparison for food dataset, question 1

	(0.7/0.3/0.7)	***(0.7/0.3/0.8)***	***(0.7/0.3/0.9)***	***(0.8/0.2/0.7)***	***(0.8/0.2/0.8)***	***(0.8/0.2/0.9)***	***(0.9/0.1/0.7)***	***(0.9/0.1/0.8)***	***(0.9/0.1/0.9)***	***Complete***
1st	*0.6642*	*0.6876*	*0.6908*	*0.6502*	*0.6652*	*0.6918*	*0.6382*	*0.6532*	*0.7097*	*0.688*
Top 5	*0.5389*	*0.5706*	*0.5886*	*0.5209*	*0.5369*	*0.5716*	*0.4989*	*0.5209*	*0.6006*	*0.5765*
Top 10	*0.5382*	*0.5515*	*0.5645*	*0.5172*	*0.5372*	*0.5495*	*0.5102*	*0.5162*	*0.5845*	*0.5705*
Top 20	*0.5187*	*0.5301*	*0.5461*	*0.5007*	*0.5217*	*0.5301*	*0.4887*	*0.5037*	*0.5501*	*0.5392*
Time	*203.5*	*185.4*	*29.7*	*26.4*	*4851.4*	*251.7*	*159.4*	*4869.4*	*31.5*	*28.1h*

Table 6.16: Cosine Similarities comparison for food dataset, question 2

	(0.7/0.3/0.7)	***(0.7/0.3/0.8)***	***(0.7/0.3/0.9)***	***(0.8/0.2/0.7)***	***(0.8/0.2/0.8)***	***(0.8/0.2/0.9)***	***(0.9/0.1/0.7)***	***(0.9/0.1/0.8)***	***(0.9/0.1/0.9)***	***Complete***
1st	*0.6652*	*0.6856*	*0.6878*	*0.6482*	*0.6632*	*0.6848*	*0.6322*	*0.6482*	*0.7047*	*0.6838*
Top 5	*0.5369*	*0.5716*	*0.5866*	*0.5169*	*0.5359*	*0.5666*	*0.4969*	*0.5149*	*0.5966*	*0.5757*
Top 10	*0.5332*	*0.5449*	*0.5605*	*0.5172*	*0.53045*	*0.5435*	*0.5042*	*0.5175*	*0.5765*	*0.5655*
Top 20	*0.5175*	*0.5251*	*0.5448*	*0.4987*	*0.5187*	*0.5231*	*0.4837*	*0.4977*	*0.5461*	*0.5342*
Time	*203.5*	*185.4*	*29.7*	*26.4*	*4851.4*	*251.7*	*159.4*	*4869.4*	*31.5*	*28.1h*

Table 6.17: Cosine Similarities comparison for food dataset, question 3

	(0.7/0.3/0.7)	***(0.7/0.3/0.8)***	***(0.7/0.3/0.9)***	***(0.8/0.2/0.7)***	***(0.8/0.2/0.8)***	***(0.8/0.2/0.9)***	***(0.9/0.1/0.7)***	***(0.9/0.1/0.8)***	***(0.9/0.1/0.9)***	***Complete***
1st	*0.5522*	*0.5686*	*0.5728*	*0.5352*	*0.5502*	*0.5738*	*0.5202*	*0.5342*	*0.5967*	*0.6718*
Top 5	*0.4219*	*0.4566*	*0.4726*	*0.4029*	*0.4239*	*0.4586*	*0.3799*	*0.4009*	*0.4826*	*0.5827*
Top 10	*0.4182*	*0.4319*	*0.4475*	*0.4032*	*0.42145*	*0.4325*	*0.3932*	*0.4035*	*0.4635*	*0.5465*
Top 20	*0.4045*	*0.4131*	*0.4328*	*0.3867*	*0.4057*	*0.4121*	*0.3727*	*0.3887*	*0.4341*	*0.5242*
Time	*203.5*	*185.4*	*29.7*	*26.4*	*4851.4*	*251.7*	*159.4*	*4869.4*	*31.5*	*28.1h*

Chapter 7

Data quality control & management

Environmental monitoring is based on time-series of data collected over long periods of time from expensive and hard to maintain in-situ sensors available only in specific areas. Climate change has prompted the monitoring of extended areas of interest and has raised the question of whether such monitoring can be complemented or replaced by low-cost and easy to use portable sensors and devices. This will allow the collection of needed information with the support of volunteers, enabling the monitoring of aquatic ecosystems and extended areas of interest.

In recent years, scientists have come to understand that widespread access to technological platforms such as smart phones and tablets, with accurate GPS trackers, high resolution cameras and multiple sensors, has opened the way to the general public to participate in scientific research [278, 71]. As the collaboration between scientists and citizens in the data collection has increased, the term citizen science has gained in popularity.

As a result, people from different social, economic and educational backgrounds are interested in contributing to the collection of data. However, such diverse participation often means that the volunteers do not have the needed experience and training to collect data in a strict scientific way, by following rules and procedures and recording in detail their actions and measurements. Data collectionis strictly defined as the process of gathering and measuring information on variables of interest, in an established and systematic fashion that enables one to answer stated research questions, test hypotheses and evaluate outcomes [252].

The deviation of the crowd-source data collection from the strict scientific definition has resulted in many scientists expressing skepticism regarding the quality of the collected data from volunteers and questioning the value that these add to their research. The main challenges regarding the use of citizen science [107, 72, 54, 248] for scientific research are

Data quality. *Critics have raised concerns about data quality, and some studies have shown controversial results. Some studies find that volunteers are inaccurate when tasked with actions that require a great degree of knowledge in a specific field, such as identifying plant species [223]. Other studies, however, indicate that while citizens have a greater learning curve than professionals, they tend to be more focused and dedicated, paying attention to details that professionals may overlook and provide data of the same or higher quality in the long run [271, 172].*

Communication between volunteers and scientists. Another challenge that materialized from the first efforts of scientists to organize data collections with the help *of volunteers was the communication between the two groups [306]. Scientists are used to speaking in scientific terms with which citizens may not be familiar or fully comprehend. This leads to the collection of data in improper ways that do not support the scientific purpose for which they were collected. Volunteers, on the other hand, tend to know better the local topology, the accessibility of the area and the local challenges.*

Bias contributions. Concerns have also been raised about the motivations of citizens and their potential biases, which might influence their activities in citizen science projects, especially for cases where the rewards were not direct [263, 180]. As many examples have shown, research in general is not devoid from bias [216]. Regardless of their origin, biased contributions should be identified and eliminated from the datasets before the data utilization.

Equipment requirements. Most technological needs for crowd-sourcing data require minimum equipment from the volunteer, a smart phone or tablet, an application and some network connection. There are cases, however, where the scientists want to measure other parameters, such as air temperature and quality, that cannot be provided by equipment already available to the volunteer. In such cases, the scientists are asked to identify sensors, that can be given to the volunteer and facilitate the data collection [116]. While such efforts have proven to be reliable and provide the needed information for the monitoring systems, occasionally, data collected by such sensors may be of low quality or affected by noise, requiring additional quality control measures.

Volunteer engagement and communities of practice. There are many difficulties regarding the engagement of volunteers and the formation of communities of practice [329]. For tasks that require regular and frequent data collections issues about recruitment, commitment and time availability may be a significant barrier. For collection of data to areas that are isolated or not easily accessible, the commute to the area may be disengaging due to the time commitment and the financial burden. In order to mobilize and maintain large citizen science initiatives, it requires the dedication of both time and financial resources. This raised the question of whether investing in citizen science can lower the cost of the data collection process. Recent studies [106] show that the answer to this question is highly dependent not only on the type of data collected but also the type and extent of errors committed by volunteers.

Data discovery and re-usability. Another important challenge regarding data collection through citizen science projects is the data discovery and re-usability. More often than not, data collected in the context of citizen science projects are very specific, measuring data in customized ways that facilitate the collection from people without scientific knowledge and are rarely mapped to generic formats or provided using widely accepted standards. The answer to this problem, comes through the utilization of the appropriate standards [240], that allow the uniform provision of data between different citizen science projects [279, 173].

Contributions. *We present here the Scent toolbox, an integrated monitoring system that aims to address all the identified challenges and provide a way for scientists to collect accurate and trustworthy environmental data. A collection of smart collaborative and innovative technologies allow citizens to support scientists and policy makers by collecting environmental measurements. The toolbox has been carefully designed to offer:*

Data quality assessment. *A data quality control mechanism has been implemented that uses a custom mechanism to identify biased or low quality contributions and remove them from the system.*

Communication between scientists and volunteers. *The Campaign Manager is a web platform dedicated to the unobstructed communication between scientists and volunteers. Scientists can use the tool to define areas and points of interest (PoIs) through an interactive map visualization and specify the type of data needed at each point of interest.*

Data Assurance. *Important measurements for the monitoring of aquatic ecosystems, such as water level and water velocity are collected indirectly. The volunteers are asked to collect images and video, which are processed by dedicated tools that extract the measurements. This eliminates biased measurements, errors due to inexperience or insufficient training and allows for a detailed post-collection scrutiny of the provided contributions.*

Integrated portable environmental sensors. *For measurements that require the utilization of additional infrastructure, small, low-cost and reliable portable sensors are selected. These provide accurate air temperature and soil moisture measurements with low maintenance needs. The application responsible for the collection of the measurements guides the volunteer and ensures the proper utilization of the sensor while a dedicated quality mechanism ensures the validity of the information.*

Volunteer engagement. *We have developed a volunteer engagement strategy that is based on gamification. The volunteers can collect points and awards by completing simple tasks, such as taking pictures and video, and contributing to the data collection process.*

Data discovery and re-usability. *Recognizing the importance of making the collected data available, all the information is modeled, stored and provisioned using the Open Geospatial Consortium (OGC) [233] Standard SensorThings API (Application Programming Interface) [183]. The filtering functionalities of the SensorThings API allow the spatiotemporal discovery of information among the collected measurements and encourage their re-usability.*

7.1 System Architecture

In order to organize data collection activities with the help of volunteers with no technical knowledge, referred to as campaigns, the key for success is to carefully identify the information needed and establish unbiased and reliable processes for its collection in a straightforward and easy way. The established processes should be simple, easy to explain and execute, independent from the knowledge and skills of the volunteer and not dependent on expensive infrastructure. They should also be designed in ways that will ensure the data assurance and provide ample opportunities for data validation and quality control.

The information, in the context of monitoring water ecosystems that has been defined as very important is the monitoring of land cover and land use (LC/LU) changes, the water level and the water velocity as well as the recording of environmental parameters such as air temperature and soil moisture. These needs have been mapped to straightforward, time-effective and meaningful actions that the volunteers can carry out with ease, without requiring special knowledge or skills. The proposed solution for the monitoring of the LC/LU changes is the collection of images of the

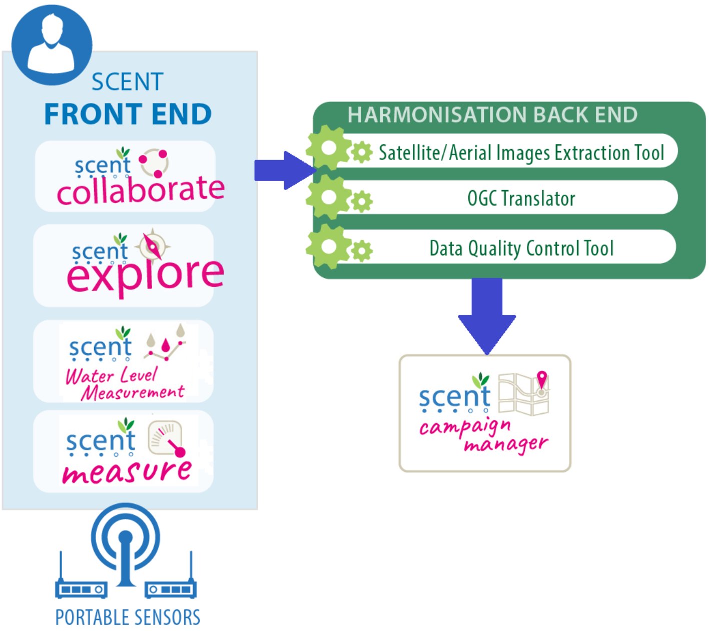


Figure 7.1: Scent toolbox architecture.

area of interest and their classification based on a taxonomy to identify depicted elements of interest. For the water level, the volunteer is again asked to capture an image, while for the water velocity the volunteer is asked to collect video. As for the collection of environmental parameters, a small, low-cost but accurate sensor paired with a smartphone or tablet is used.

Aiming to facilitate the users in performing these tasks and address the abovementioned challenges related to the data collection through volunteers, we have designed an innovative system, which is presented in Figure 7.1. The system is composed of a suit of components, each one specifically designed to support the tasks assigned to volunteers or to facilitate the data exchange. An innovative flow of information has been implemented to ensure that the needs of the local authorities are communicated to volunteers, while the collected data are forwarded to environmental experts where they are processed and enriched before being presented to the authorities that can now have improved monitoring of the area of interest. The system requirements for the toolbox have been collected through multiple end-user workshops and focus groups, as discussed in [307]. The key components of the system are presented in detail below.

7.1.1 LC/LU Taxonomy

When monitoring an ecosystem, it is very important to have detailed, area specific, information about the land cover and land use. Indicative of the importance of such monitoring are the efforts dedicated by the European Union to the development of a consistent land cover repository. The Copernicus Land Monitoring Service [55] utilizes the CORINE (Coordination of Information on the Environment) Land Cover (CLC) inventory, which was initiated in 1985 and updated in 2000, 2006, 2012 and 2018. It consists of an inventory of land cover in 44 classes. CLC uses a Minimum Mapping Unit (MMU) of 25 hectares (ha) for spatial phenomena and a minimum width of 100 m for linear phenomena. CLC is produced by the majority

Table 7.1: Taxonomy Elements.

Urban Case—LC/LU	**Obstacles or Damages**	**Rural Case—LC/LU**
Buildings	Trees	Furrows
Paved areas	Tree branches	Rice fields
River bank. Low grass	Cars/vehicles	Vineyards
River bank. Shrubs	Dustbins	Fruit trees
River bank. Concrete	Storm drains. Clean	Pastures/Grassland
River bank. Stone	Storm drains. Debris	Forest. Broad-leaved
River bank. Bare soil	Storm drains. Leaves	Forest. Coniferous
River. Dry cross-section	Storm drain discharge	Forest. Mixed
Roads	Buildings. Damaged	Heathland
Parking lots	Roads. Damaged	Sand
Railway lines		Bare rock
Excavation or construction sites		Sparsely vegetated area
Dump sites		Burnt area
Parks. Tall vegetation		Inland marshes

of countries by visual interpretation of high resolution satellite imagery. In a few countries semi-automatic solutions are applied, using national in-situ data, satellite image processing, GIS integration and generalization. The 44 classes that form the CORINE Taxonomy [56] come from a 3-level hierarchical classification system that respects the detail level of the 25 ha MMU. As an example, for the classification of Water bodies, a level-1 class, there are available two sub-classes, Inland waters and Marine waters. The Inland waters are further divided in Water courses and Water bodies.

In the context of monitoring water ecosystems, however, the CORINE taxonomy does not offer the level of detail required. Specifically, it is of utmost importance to specialize between different types of water courses, wetlands and sparsely vegetated areas. For this reason, the Water bodies class was further specified to include Dry cross section and River bank. Next the River bank was further specified to include types of land cover at the bank, such as stone, bare soil, concrete and shrubs. All the taxonomy classes used by the Scent toolbox are presented in the Table 7.1.

7.1.2 Campaign Manager

Aiming to alleviate the communication gap between scientists and volunteers, a web-based application, the Campaign Manager, was created. The Campaign Manager allows scientists to communicate their needs regarding data collection activities in a user friendly, simple and functional way. Scientists can use a user-friendly web interface to create campaigns for areas where data on LC/LU, soil conditions and river parameters are needed, while also providing descriptive information about the scope, aim and purpose of the campaigns. In addition, the scientists define for each campaign Points of Interest (PoIs), specific coordinates where data should be collected. Furthermore, the Campaign Manager supports the visualization of citizen generated data, images, videos and sensor measurements, as well as maps of the areas of interest with information regarding LC/LU.

The architecture of the Campaign Manager has been designed according to the

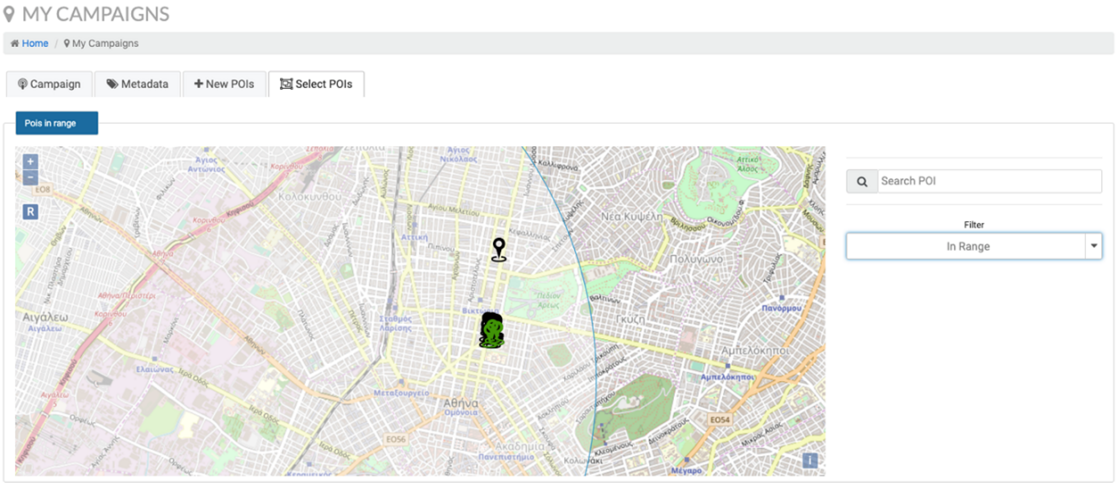


Figure 7.2: Campaign Manager. PoI visualization.

classical multi-tier system with a communication paradigm based on open source service-oriented architecture, where each service interacts with the other through a set of messages written in a standard format. The architecture and relevant services are composed of the following layers:

Presentation layer. Mobile and web interfaces, the main gateway for interacting with the application's main services.

Service layer. Services responsible for the communication between parts of the Presentation Layer and also for the communication between Campaign Manager and external API endpoints, further details about these endpoints are provided in Section 7.4.3, for the acquisition of citizen generated data.

Data access layer. The internal endpoints of the Campaign Manager.

Database. A database and collections implemented using MongoDB.

The interfaces of the application, Figure 7.2, are implemented using Angular 4, a JavaScript-based open-source platform that makes it easy to build applications with the web. Angular combines declarative templates, dependency injection, end to end tooling and integrated best practices to solve development challenges. Moreover, spatial visualizations of the collected data are implemented as map overlays using OpenLayers library along with OpenStreetMaps.

The server and the RESTful API of the Campaign Manager are implemented using Node.js, an open-source, cross-platform JavaScript run-time environment that executes JavaScript code server-side. Through the API basic user actions, such as authentication and password change, and more complicated ones, such as data CRUD (Create, Read, Update and Delete) actions, are handled. The personalized settings of the user accounts and settings are stored in a database implemented using MongoDB, a free and open-source cross-platform document-oriented database program, that uses JSON-like documents with schemas.

7.1.3 Gaming Applications

One of the major challenges in citizen science is keeping the users motivated to contribute over a period of time, in a way that will not require the continuous dedication of time and resources from the scientists. The 1% rule of thumb [227], which is also referred to as the 90-9-1 principle, aims to quantify the imbalance that is *associated with the contributions of volunteers to crowd-sourcing efforts. It has been observed that approximately 1% of users of any application or website contribute the vast majority of new content. The percentage that they contribute is calculated in most cases between 90% to 95%. An additional 9% of users are credited with sparse contributions or secondary tasks that may not be as important, depending a lot on the context discussed. An indicative example is Wikipedia where approximately 1% of users are contributing the majority of the content while 9% of users are usually editing, proofreading and correcting grammatical errors [331]. Finally, the majority of individuals, the remaining 90%, are only interested in using the available material without providing data or contributing to the overall efforts. In one of the empirical studies of this phenomenon, this observation was supported within the digital health social network context [207].*

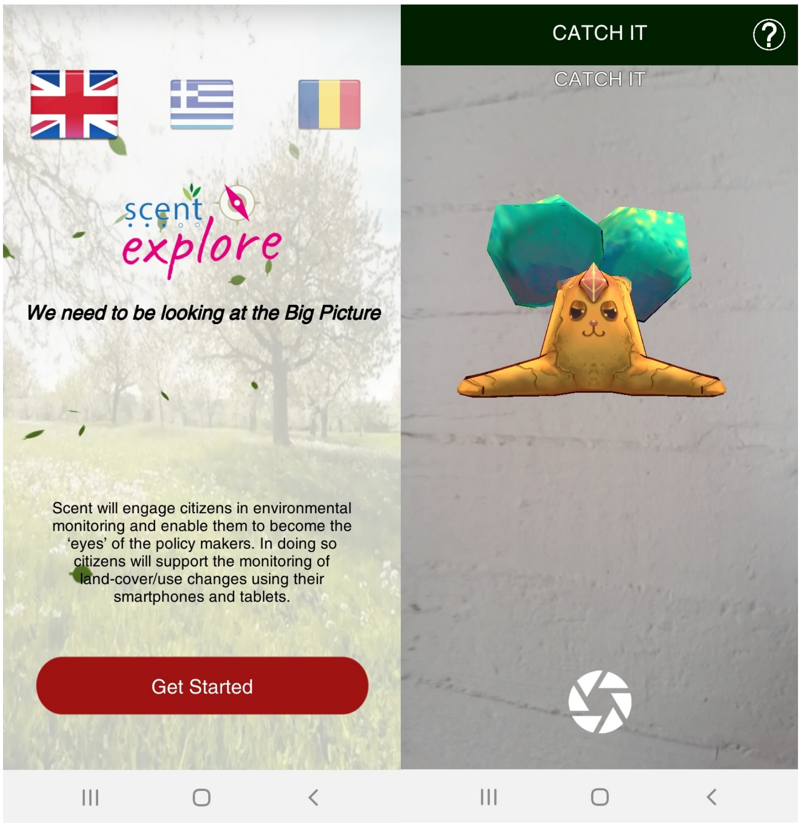
An approach to boost volunteer participation and user engagement comes through the gamification of the process [266, 145, 224]. The two mobile applications that were developed to utilize elements of gamification [51] but also support the data collection for the monitoring of the LC/LU and the collection of river parameters, are presented here.

Scent Explore

Scent Explore, Figure 7.3, is a game designed for the primary purpose of collecting scientific data and not for pure entertainment that the volunteers use to locate the PoIs, as defined by scientists through the Campaign Manager; carry out the tasks specified and gain rewards when successfully concluding them.

First, volunteers select a campaign and see the associated PoIs and the actions to be performed. As the volunteer reaches a PoI where an image of LC/LU is needed, the phone camera is activated and features of augmented reality, a little animal, are shown; see Figure 7.3. The volunteer is encouraged to look around and locate the little animal. While the volunteer is looking for the AR feature, the application is integrating the information received from the GPS with the gyroscope and the accelerometer so as to guide the user to the correct position, the one provided by the local authorities at the Campaign Manager. The little animal is carefully integrated to the screen so as to appear further away or close depending on the distance of the volunteer from the PoI. For older devices that lack some or all of the needed sensors for the navigation to the PoI, there are additional features that ensure an unobstructed user experience and valid collection of data. After locating the augmented reality feature, the volunteer can simply tap on the screen to capture it and as a result take an image with specific orientation that includes the information of interest [103, 268]. The volunteer is then asked to choose elements of the taxonomy that best describe the content of the image. Upon the conclusion of this task, the user is rewarded by adding the captured little animal to their collection and with points depending on the distance from the PoI. Collecting points and badges, based on the level of expertise, is a common way of user engagement for gaming applications that has been proven to have positive results [75, 102].

When a volunteer reaches a PoI where the water level measurement is required, the process described above is repeated, only this time the augmented reality elements are strategically located towards the water level indicator. Additional information about the image processing, its requirements and the measurement extraction are given in Section 7.1.4. For a PoI where the water velocity is required, the camera of



We need to be looking at the Big Picture

Scent will engage citizens in environmental monitoring and enable them to become the 'eyes' of the policy makers. In doing so citizens will support the monitoring of land-cover/use changes using their smartphones and tablets.

Figure 7.3: Scent Explore.

the device is activated to video mode. The volunteer is then asked to capture a video. Additional information about the requirements for the video and the extraction of the water surface velocity are provided in Section 7.1.4.

An additional feature of Scent Explore, aiming to improve the user experience and support user engagement, is the random PoI generator. In case that two PoIs, as defined in the Campaign Manager, have a significant distance or the user is delayed in any other way, e.g., people in the team taking longer to complete the PoI resulting in the user being inactive for more than three minutes, then the application creates a intermediate PoI requesting LC/LU information. This feature not only succeeds in keeping the volunteer engaged in case of delays during the campaign but also results in the collection of additional data.

Given that the collected video and images are further processed to extract river measurements, as discussed in Section 7.1.4, it is important they are of high quality. To this end, further actions are taken to ensure the quality of the collected multimedia. To begin with, images are stabilized by directly accessing the charge-coupled device camera sensor, improving the color schema and eliminating noise from movement, instead of using pictures interpolation. Image stabilization is additionally improved by taking three photos every time that a volunteer is capturing an augmented reality feature, rather than the expected one, and choosing the photo with less variation of the mobile accelerometer, to ensure better definition of the captured elements. Furthermore, the video stabilization is achieved by utilizing the accelerometer to detect instability and up-down movements at each frame. Additionally, extra thought is given to the positional data quality. The application gets GPS data with an implementation of Position Dilution of Precision (PDOP). This is very impor-

tant given that with the GPS, optimal accuracy is in the order of 3–5 m or 10–20 m in isolated environments; with PDOP, the position accuracy can be less than 2 m or 3–5 m, respectively.

Scent Measure

The first step for the Scent Measure was to identify a set of requirements to be used in order to identify the proper sensor for the volunteers to use in the field. The requirements take into consideration the needed volume, the inexperience of the volunteers as well as the need for an interoperable and reproducible solution for different pilot areas. The identified requirements are presented in Table 7.2.

After an exhaustive search for the available sensors that can measure air temperature and soil moisture, the Xiaomi Flower Care Smart Monitor [335], as presented in Figure 7.4, was selected as it complies with all the above mentioned requirements. It records measurements quickly and is simple to use as it does not require any installation. It provides measurements for air temperature and soil moisture every 15 s requiring only to insert the metal probes into the soil. It is a reliable sensor, with a battery life of a year and resistant to corrosion with a waterproof battery compartment. It is lightweight, only 130 g, and compact, only 12.00 × 2.45 × 1.25 cm in dimensions, making it portable. The detectable temperature range covers from −20 °C to 50 °C with ±0.5 °C temperature error control. The moisture is measured in scale 0%–100% with 1% accuracy.

Scent Measure, shown in Figure 7.5, is the application dedicated to the collection of sensor measurements by the volunteers. The application can be used independently, in areas where only sensor measurements are needed or complimentary to Scent Explore. The users can use their Scent account with this application too, in order to collect points and rewards for their contributions. In order to maintain the same look and feel among the applications, the user is rewarded with points for each collected measurement. In addition, badges dedicated to the collection of sensor measurement are available and unlocked based on the performance of the user.

The main data flow of the application is to connect to the portable sensor of the volunteer, collect measurements, add the needed metadata and forward them to the dedicated backend server. In detail, first the user can choose to login or play as a guest. Next, the application scans for nearby sensors. For communication with the portable sensor, the application uses Bluetooth Low Energy as it is designed to connect devices with low power consumption. All the discovered sensors are shown to the user so as to select the one that is assigned and proceed with the measurement. The application then collects air temperature and soil moisture values at 15*secs time intervals from the selected sensor; the GPS coordinates and the current time as it is available at the mobile device are added as metadata to the measurement which is then forwarded to the backend server. To further support user engagement, the application notifies the user every time a new measurement is recorded and displays the recently collected measurements.*

7.1.4 River Measurement Extraction from Multimedia

The collection of measurements from water bodies, such as rivers and canals, are usually carried out by groups of scientists trained to accurately collect such measurements without bias, using a variety of methods to ensure quality control [319, 289,

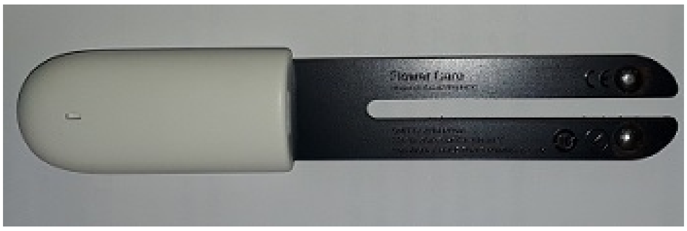


Figure 7.4: Xiaomi Flower Care Smart Monitor.

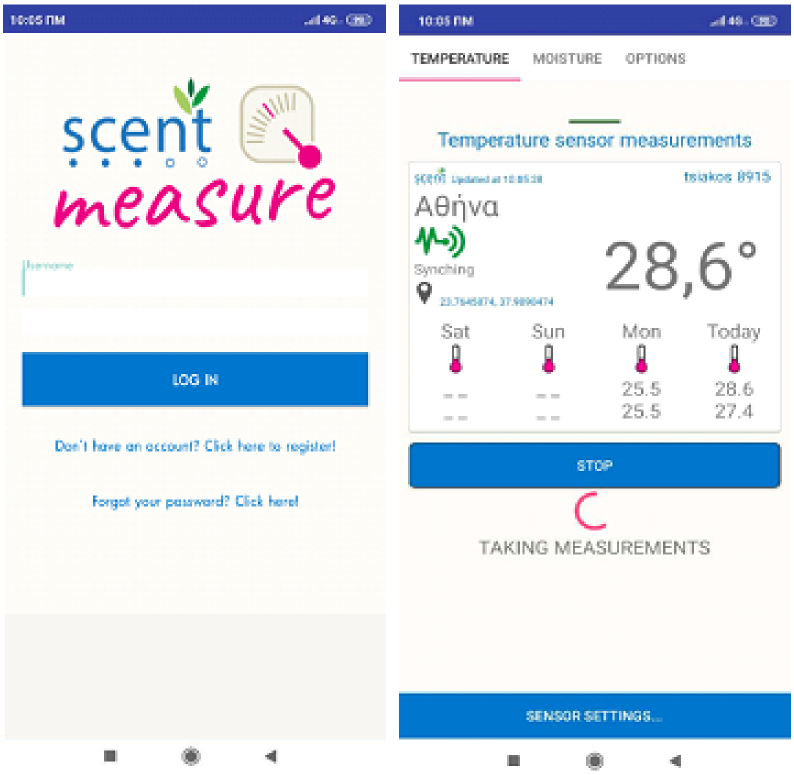


Figure 7.5: Scent Measure.

337]. Such data collections require expensive equipment and are time consuming. The utilization of such techniques in the context of citizen science is not feasible as they are often complex, require training and resources and lead to variable degrees of data quality based on the expertise of the volunteer contributing.

In order to provide measurements for the proper monitoring of ecosystems, the data should be consistent, accurate and independent of the expertise of the volunteers collecting it. To achieve this, volunteers are simply asked to take a picture of a water level indicator or a video of a predefined floating object and upload it through the Scent Explore. The crowd-sourced data are then processed by two tools, that have been specifically designed to automatically extract the water level and the water velocity in a consistent and accurate way.

Water Level Measurement Tool

The WLMT receives an image that contains a water level indicator and the waterline, the line where the water level indicator meets the surface of the water. Initially, the tool uses the color difference in the image to identify the waterline. This step is very

Table 7.2: Requirements for portable sensor.

Requirement	Justification
Measurements to be recorded.	According to the requirements of the environmental scientists, the key parameters that the portable sensors should be able to record are air temperature and soil moisture.
Low-cost.	As many sensors as possible are needed in the field simultaneously; where it is probable they might get destroyed or lost, the price per unit for the sensor should be low. This, however, should not compromise the quality and accuracy of the collected measurements.
Portability.	Keeping in mind that the volunteers will have to carry the sensors for the duration of the Campaign, there is a need for the sensor to be lightweight. In addition, it is crucial to ensure that the sensor will have enough power autonomy for at least one fully operational day, to avoid the need for power banks or charging stations during the Campaign.
Ease-of-use.	Taking into consideration the inexperience of the volunteers, their lack of training regarding the scientific way for the collection of environmental measurements, the possible challenges on the field and the need for multiple measurements in a short time span, the portable sensor should have a very intuitive way of use and require minimum effort from the user.
Connectivity.	Aiming to have a high degree of accuracy and ensure uniformity among the measurements collected, many of the needed values such as the GPS coordinates and the timestamp are collected from the mobile device of the volunteer. This means that the sensor should be able to connect, preferably over Bluetooth, with a mobile device and forward the information there.
Open API.	Last but not least, the collected information is to be forwarded to the system backend. In order for that to be feasible the sensor should provide an open API.

important for the detection of the correct measurement, the numbers closer to the waterline, and eliminates any noise from numbers of the water level indicator higher up or random signage included in the image.

Next, the tool "reads" the indicator and extracts the number that is closest to the waterline. The conversion of images containing handwritten or printed text into machine encoded text is called Optical Character Recognition or Optical Character Reader (OCR). Tesseract [281] is an optical recognition engine, released under the Apache License, Version 2.0, and sponsored by Google since 2006 [320]. Tesseract is considered one of the most accurate open-source OCR engines available [257, 332]. In our case, a Python script utilizes this library to extract the water level.

While the digit extraction is very accurate and robust, the main issue is the differentiation between water level indications, specifically if a topological or a hydrological indicator is captured in the image. In the majority of the campaigns, hydrological indicators were used, but in some cases during the first campaigns where the issue was not known, some topological indicators were used as well. This is an important problem as the two indicators display the numbers very differently. The hydrological indicators use centimeters while the topological indicators use decimeters, so an

Table 7.3: Instructions for the water level indicator capture.

Step Number	**Description**
#**1**	Locate the water level indicator at the opposite river bank.
#**2**	Wait for the camera to be activated.
#**3**	Stand straight opposite to the water level indicator.
#**4**	Chose the camera frame to include the water level indicator and the waterline.
#**5**	Make sure that you keep the camera stable and steady.
#**6**	Capture the image.

identified value of 10 *corresponds to 10 cm for a hydrological indicator but to 10 dm for a topological one.*

Aiming to ensure measurements with high degree of confidence, detailed instructions are given to the volunteers, as presented in Table 7.3, regarding the optimal way that they are to capture images including a water level indicator.

Water Velocity Calculation Tool

The WVCT receives a video with a pre-defined object, which is floating downstream following the river course, and extracts the water surface velocity. As a floating object, a tennis ball was selected. Initially, efforts were dedicated to more eco-friendly solutions such as oranges. Such solutions however pose significant technical challenges, given that each object is expected to have different color and size and differ in density. Upon testing, it was observed that occasionally oranges were submerged into the water rather than floating on the water surface. Tennis balls have strict specifications, they are always the same bright yellow color, have a 6*cm diameter and float on the water surface. To ensure that no tennis balls will pollute the ecosystem, a very thin fishing string was carefully attached to the tennis balls used during the campaigns to ensure that the ball would be always retrieved.*

The main challenge regarding the processing of the video is the noise that has been introduced to the recording due to intentional or unintentional movement of the mobile device. In order to eliminate this noise the video is first stabilized. The stabilized video is then processed frame by frame. The first step is to locate the first frame of the video that contains the tennis ball. This is achieved by looking for the very distinct bright yellow color of the tennis ball, which is not expected to occur naturally in the environment. Each pair of consecutive frames are examined, and the optical flow between them is calculated. First, the average optical flow of the area of the frame that includes the tennis ball is calculated. Next, the average optical flow of the rest of frame is calculated to identify if the camera was moving during the video capture and eliminate this movement from the calculation. The displacement of the ball, between these two frames, is calculated by subtracting from the average optical flow of the ball the average optical flow of the rest of frame.

The calculation is repeated as long as the next video frame contains the tennis ball. A dedicated counter is recording the number of frames that were valid and examined while the total displacement is calculated as the sum of the displacements between all pairs. The end of the video is defined as the first frame where a tennis ball cannot be

Table 7.4: Instructions for video recording.

Step Number	**Description**
#**1**	Select an area of the river where there are no visible obstacles and/or river bends.
#**2**	Activate the camera and chose the appropriate frame.
#**3**	Throw the securely attached tennis ball upstream.
#**4**	Start the video recording while waiting for the object to float by.
#**5**	Record the object floating by while keeping the camera stable and steady.
#**6**	When the object is no longer visible within the camera frame stop the video recording.
#**7**	Retrieve the tennis ball using the fishing line.

located. This is an additional safety measure in the case where excessive movement of the camera resulted in the tennis ball being completely removed from the frame. The total displacement, which has been calculated in pixels, is then calibrated using the dimension of the tennis ball given, which we know to be 6 cm in diameter. Finally, the number of frames where the tennis ball is found is calibrated by the frames per second of the video, to give the time in seconds calculated for the displacement. We calculate the water velocity by dividing the total displacement by this time resulting in the measurement of the water surface velocity in cm/s.

The tool has some baseline validation mechanisms that are used to discharge video that is of low quality. The first validation test is with regard to the duration of the presence of the tennis ball. If the tennis ball is tracked in fewer frames than the frames per second of the video, giving less than a second of useful material, then the video is considered invalid. The second validation test examines the calculated displacement of the tennis ball. If the displacement is less than the identified size of the tennis ball, then the video is again considered invalid. In any other case, the water velocity is extracted with a confidence score that it calculates as the average of the percentage of the video frames that were used, continuous identification of the tennis ball, to the total duration of the video and the average number of pixels that the tennis ball covers in the frames to the overall displacement as calculated in pixels. The function giving the confidence level is:

$$ConfidenceLevel = \frac{1}{2}(\frac{ValidVideoFrames}{TotalVideoFrames} + \frac{TennisBallSize}{Displacement})$$

The tool was tested with some videos captured for this purpose, containing frames of the tennis ball. The tracking of the object was very reliable and robust to the video quality and duration; the displacement calculation however was up to some degree dependent on the stability of the video. Aiming to ensure measurements with high degree of confidence, detailed instructions are given to the volunteers, as presented in Table 7.4, regarding the optimal way that they are to capture the video.

7.2 Demonstration Cities and Data Collection

The Scent Toolbox was tested in two carefully selected areas, the Danube Delta in Romania and the Kifisos River basin in Greece. The areas offer different environments and topologies and have different needs and challenges allowing the testing of the system under different conditions. Starting from August 2018, several specific campaigns were organized for each study area. Each campaign was focused on one of the needed measurements in order to facilitate a training workshop before each one. During the workshop, the volunteers were informed about the project and trained in the use of the Scent Toolbox component that was used during the campaign. The campaigns ran for a period of 10 months, with 6 different campaigns taking place in each pilot area, two for each thematic topic, LC/LU images, sensor measurements and river data.

7.2.1 Danube Delta, Tulcea, Romania

The Danube Delta in Tulcea, Romania [69] is the largest natural wetland in Europe and is protected under UNESCO as a unique biosphere reserve. The Danube Delta Biosphere Reserve has the third largest biodiversity in the world with more than 5500 *species of flora and fauna and the highest concentration of bird colonies in Europe. Many of the species of plants and animals living in the Danube Delta are unique to it. The Biosphere Reserve is also home to a population of more than 10,000 people, who rely on the natural resources and the environment of the Danube Delta for their livelihoods.*

However, the Delta has suffered from human interventions, which led to dramatic changes in the area, including damming large areas for agricultural use, fishing and forestry. This has resulted in the disturbance of the ecological balance of the wetlands and in some cases the deterioration and loss of biodiversity. Monitoring the changing landscape of the Delta is an important first step in maintaining its natural ecological balance, and protecting its communities. The Scent Toolbox aims to collect up-to-date data to dynamically and accurately monitor changes in land-cover and land-use and thereby help to better protect both the Delta's environment and the population who live there.

Campaigns in Numbers *During the campaigns* 200 *participants collected* 1500 *videos,* 1500 *sensor measurements, 1800 images containing water level indicators and 16,000 images with LC/LU elements.*

7.2.2 Kifisos River Basin, Attica, Greece

The Kifisos river basin[259] is roughly 380 km^2, and almost 60% of its watershed is urbanized as the metropolitan area of Athens. As the city has expanded, the land-cover of the area has transitioned from rural to urban, and industrial in some areas. The hydrological network of the basin has been heavily engineered to support expanding constructions. However, in many cases, the hydraulic works were poorly designed. There are many areas where there are illegal constructions, even within the main river course. As a result, during periods of heavy and rapid rain events, the river floods due to the insufficiency of drainage networks, causing severe damage to infrastructure around the river.

Scent has organized campaigns in areas of the river basin characterized by rapid changes in the river water level caused by natural and anthropogenic factors. With the Scent Toolbox, volunteers collected important information that allows for increased awareness of the status of the river environment. The volunteers have also collected soil moisture and temperature measurements, both before and after heavy rain events to help hydrologists better understand the river dynamics.

Campaigns in Numbers *During the campaigns* 460 *participants collected* 400 *videos,* 1000 *sensor measurements,* 700 *images containing water level indicators and* 4000 *images with LC/LU elements.*

7.2.3 Safety Considerations during Campaigns

For the campaigns organized at the Danube Delta, personnel from the DDNI [64] and SOR [284] were responsible for ensuring the safety of the volunteers. Both organizations are partners of the Scent project and experienced in organizing campaigns for educational or bird watching purposes in the area that was visited. In all cases, safety precautions were taken, including life vests for boat trips, protection from the weather elements such as hats and sunscreen in the summer and raincoats in the winter as well as satellite and radio communications in case of an emergency while at remote areas without GSM (Global System for Mobile Communications) signal.

For the campaigns organized at the Kifisos river in Attica the Hellenic Rescue Team of Attica (HRTA) [140], a Scent project partner was responsible for the safety of the volunteers. The HRTA members have significant experience in providing first-response emergency medicine in events with many participants as needed. For most of the campaigns organized at the Kifisos river, pedestrian paths in semi-rural environment were chosen, so the HRTA members were mostly tasked with coordinating the teams of volunteers and ensuring that all safety guidelines were respected. As an exception to this, due to the topological challenges of an area of interest at the northern part of the Kifisos river, only trained HRTA members were allowed to participate.

7.3 Data Quality Control

7.3.1 LC/LU Annotations

Volunteers are given training over the available taxonomy elements as well as examples of images to which these correspond. This training process provides the relevant knowledge to the volunteers in order to provide annotations of high accuracy. The environment of the campaign, the high inter class variation of the taxonomy and the unpredictability of the environment may lead the volunteer to provide incorrect or misleading annotations. In order to ensure that such annotations will be removed from the dataset, a validation mechanism has been implemented.

Scent Collaborate is a browser-based crowd-sourcing platform that allows users to validate images collected and annotated during the campaigns. The users can also provide further annotations for elements that are in the images but have not been included in the annotations. Each image is annotated by one or multiple users, depending on whether there is an agreement or not upon the validity of the annotation. The validity of the annotation is established based on the majority vote.

Aiming to boost user engagement and the user-friendliness of the toolbox, the user can login to the Scent Collaborate using the Scent account. Then the user is rewarded with points based on the number of annotated pictures. Once more, the points are connected to badges that acknowledge the achievements of the users. Scent Collaborate is designed to engage users that do not live close to any of the study areas but are still interested in contributing to the Scent movement.

7.3.2 River Data Measurements

Understanding that due to low quality contributions and calculation inaccuracies, the measurements extracted by the WLMT and the WVCT tools can contain invalid data, a methodology has been developed that identifies such values and removes them from the datasets. We present here the steps of the methodology as well as the results of the data quality analysis for the data collected during the first Kifisos campaign, held from the 15th to the 17th of November, 2018.

The first step for the evaluation of the data extracted by the WLMT is the spatial and temporal clustering. To this end, the coordinates and timestamps available from the metadata were used to divide the measurements into K clusters using the K-means algorithm [98, 190]. The main challenge of the k-clustering process is to correctly identify the number K of the clusters into which the data should be divided. While knowing the nature of the data and the collection process can provide an intuitive way of specifying the number of clusters, we chose to use a more robust and agnostic approach, the average silhouette method [163]. Applying this step to the first Kifisos campaign, the method indicated that the optimal number of clusters is $K = 10$*; Figure 7.6. Further examining the clustered data, a internal variance was observed along with data that can be considered as outliers. In Figure 7.7 at the top, the internal variance of the clusters is presented using the box-plot technique. The next step of our methodology was to eliminate the internal variance presented in the clusters. To achieve that, the "sigma test" was applied [298].*

This test checks if data within the same cluster follow the normal distribution $N(\mu, \sigma^2)$*, and the mean value and standard deviation can be approximated by the characteristics of the sample,* $\mu \approx \bar{x}$ *and* $\sigma^2 \approx s^2$*. The sigma test is used to remove possible invalid measurements by keeping only the data in the interval* $(\mu - c \cdot \sigma, \mu + c \cdot \sigma)$*, with* $c \in \Re^+$*.*

For the data extracted during the Kifisos campaign, and for a value $c = 1$*, about 32% of data of each cluster were invalidated. The box-plots of the clusters after the sigma test are presented in Figure 7.7 at the bottom. Carefully examining the results, it is noticeable that some variance is still within the results. In order to identify the source of this variance, the outliers were visually inspected. This showed that occasionally the WLMT gave a wrong measurement. In most cases, the WLMT gave the proper reading based on the content of the image but that was significantly higher than expected as the bottom of the image did not contain the water level, or the water level indicator was obstructed by brought material. In a few limited cases, deterioration of the letters printed on the indicators made some digits resemble others such as a "7" resembling a "1" or an "8" resembling a "0".*

For the evaluation of measurements extracted by the WVCT, the same methodology was followed. In Figure 7.8, the box-plots that show the range of the collected data of the $K = 12$ *clusters for the Kifisos campaign are present. The number of*

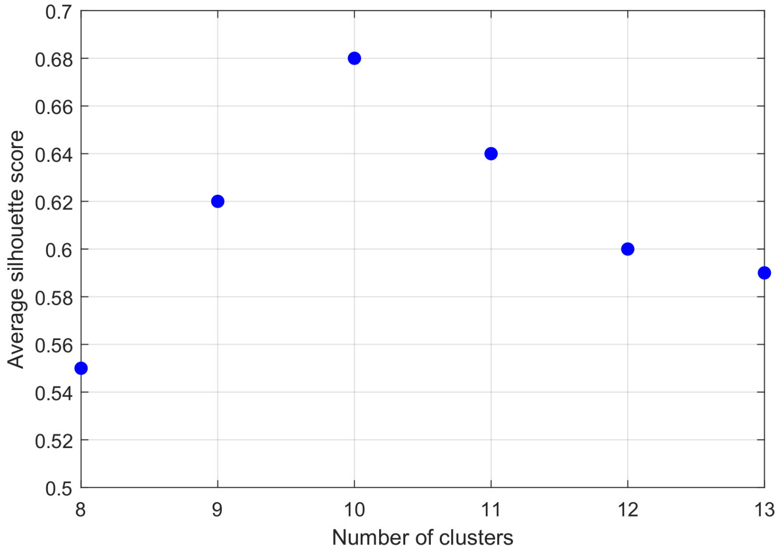


Figure 7.6: Using the silhouette method to define number of clusters.

clusters was once more decided using the average silhouette method. In some cases, there are clear outliers; these were examined further to identify the causes. The main cause was that often part of the video contained the retrieval of the tennis ball, a process that was increasing the average velocity calculated. In order to eliminate this problem, instructions were given to the volunteers and an option was added to the interface of the Scent Explore application allowing users to discard video that was not properly captured.

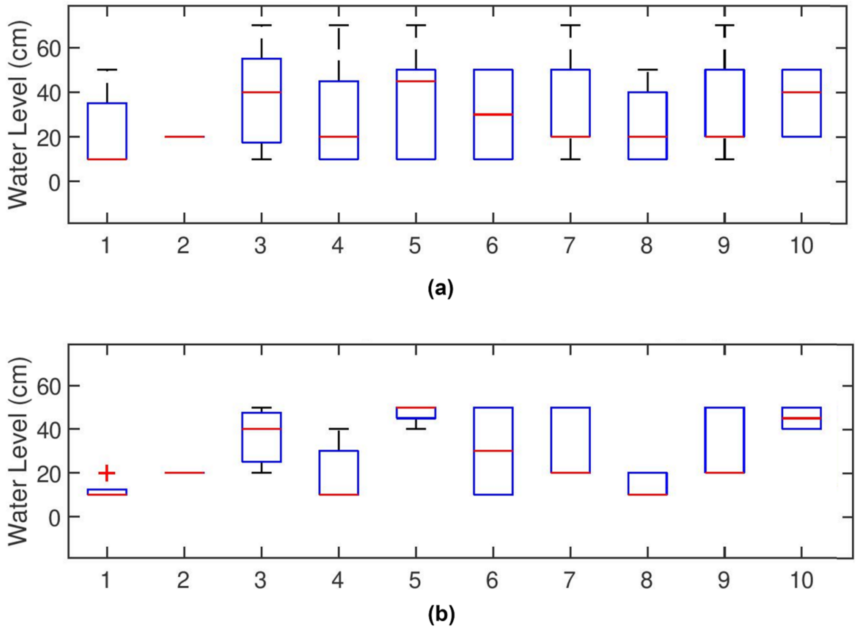
7.3.3 Sensor Measurements

Sensor measurements are accompanied by rich metadata that include coordinates, timestamp, a unique measurement ID, the unique identifier of the volunteer that collected the measurement as well as the GPS accuracy. The measurements contain two values, one for air-temperature measured in Celsius degrees and one for soil moisture as a percentage. The collected data of each campaign are examined to identify potential invalid measurements and remove outliers. For both of the collected measurements, the criteria used to identify invalid measurement are

- *Spatially isolated measurement. Knowing that volunteers collect measurements in groups of at least three persons, having a measurement that is more than 10 m away from the rest is an indication of either invalid coordinates or a contribution far from the PoI.*

- *Volunteer exclusive measurements. This test is also based in the knowledge of how campaigns are organized, in groups of people. If there is an area where only one volunteer is providing measurements, most probably there was an issue with the provided coordinates or the execution of the measurement collection process.*

- *GPS accuracy being greater than 20 m. This test is necessary, as the areas the volunteers visited were remote, and in many cases the GPS coordinates were not accurate. Given the importance of the spatial distribution for the usage of the collected measurements, such values are not useful.*



(a)

(b)

Figure 7.7: The variance of water level measurements (**a**) before and (**b**) after the sigma test.

For the air temperature measurements, an additional validation test, called the range test, is carried out. The validate range is calculated as $\bar{T}\pm$ *10 °C, where* $\bar{T}$ *is the daily mean temperature the volunteers provided. For the soil moisture measurements, the delta test [298] is applied. The analysis of the soil moisture measurements, revealed many entries with zero as value. Extensive laboratory testing proved that the sensor can provide a valid measurement of 0% moisture level when measuring dry soil without any roots of living plants inside, which was also expected by its specifications. This created the need for differentiating between measurements taken with the sensor not inserted properly to the ground thus being invalid and measurements taken where the soil was dry. In order to identify valid zero measurements, the time series of measurements provided by a user are examined. If a 0% moisture level measurement is followed within the next 15 s, which is the update rate of the sensor, with a measurement greater than 0%, then the first measurement is invalidated.*

For the Kifisos campaign, there were $N = 215$ *measurements of air temperature and soil moisture collected. Following the methodology described before, they were clustered into* $K = 4$ *clusters. The spatial distribution of the measurements collected is presented in Figure 7.9, showing that the choice of creating four clusters was accurate. The internal cluster distribution of the air temperature is uniform, distributed inside its range in most cases. However, there are cases where the distribution of the values within a cluster is irregular; such an example is presented in Figure 7.10 where the data distributions within two clusters are compared. This raises the question of a user providing systematically invalid measurements.*

In order to determine that, the data of the cluster are further divided using the unique user identifier, as shown in Figure 7.11, and the one-way ANOVA statistical test [328, 338] is applied as shown in Table 7.5. This test proves that there is

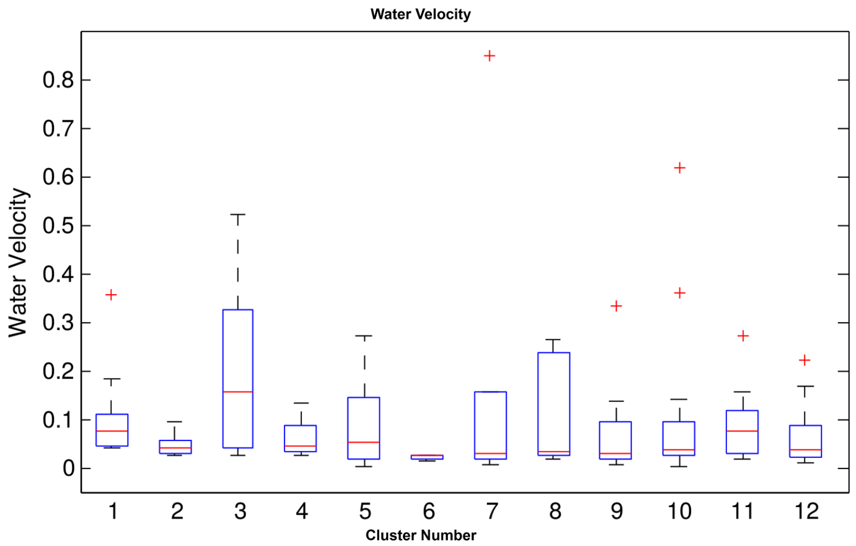


Figure 7.8: The water velocity data divided in $K = 12$ clusters.

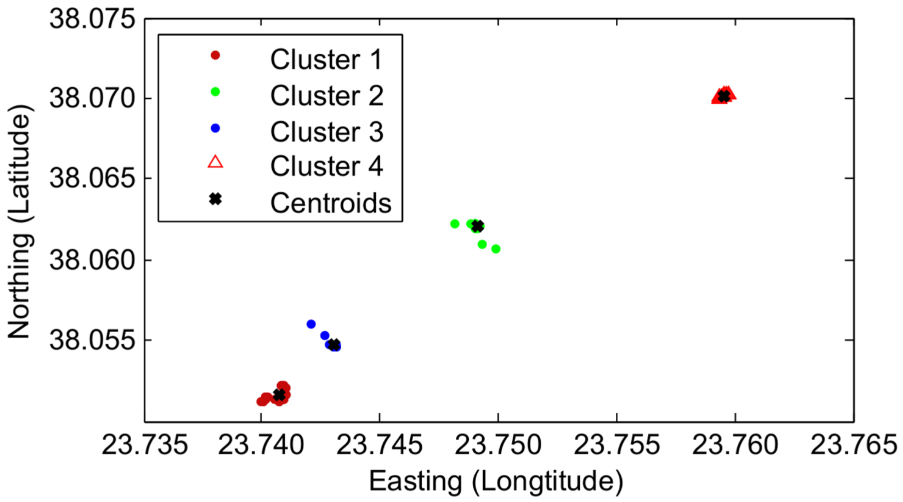


Figure 7.9: Spatial distribution of the sensor data clusters.

significant difference between measurements collected by different users within the cluster, as $p = 6.47 \times 10^{-34} < 0.01$. *The mean air temperature values showed that the user with unique identifier 368 consistently recorded higher temperatures.*

7.4 Data Provision

Taking into consideration the large volume of collected data, the need of the environmental scientists and local authorities for information based on spatial and temporal criteria as well as the importance of interoperability and data discovery, the OGC SensorThings API [183] is implemented as the storage and retrieval platform for all measurements collected. The SensorThings API is an OGC standard providing an open and unified framework to interconnect sensing devices with data and applications. It follows REST principles, the JSON encoding, and the OASIS OData protocol and URL conventions.

The foundation of the SensorThings API is its data model that is based on the

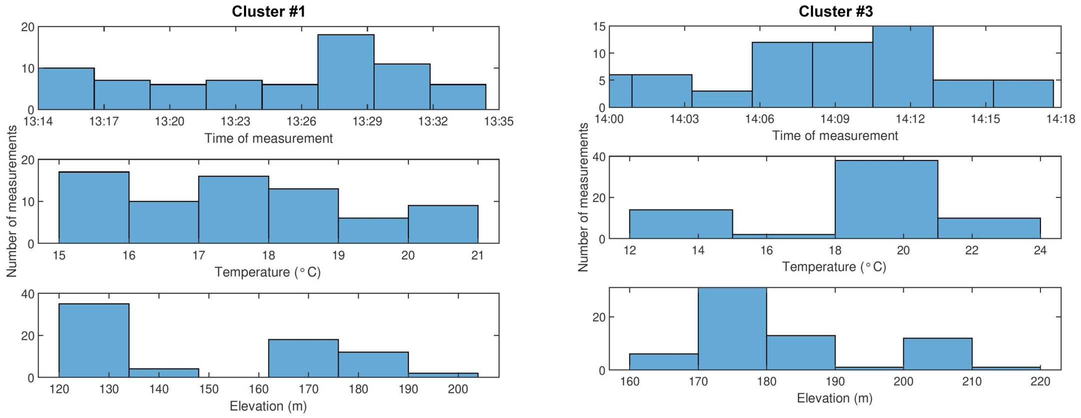


Figure 7.10: Normal vs irregular intra-cluster data distribution.

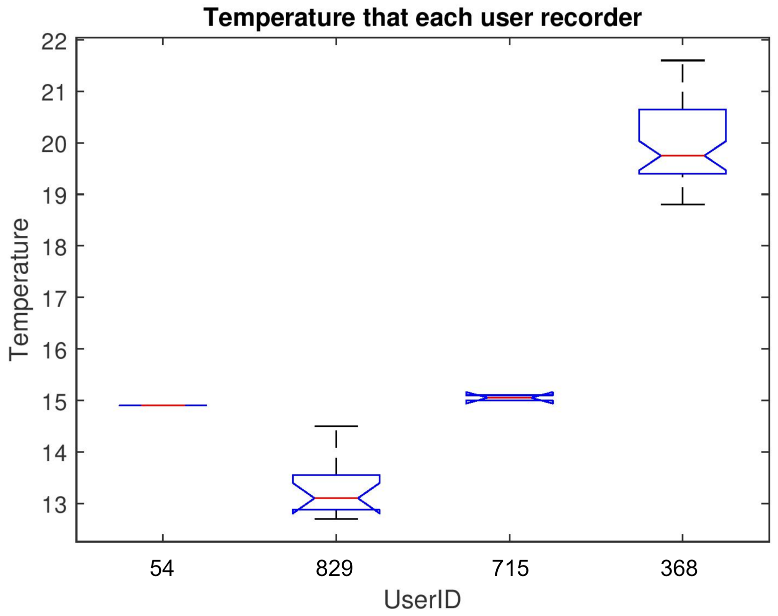


Figure 7.11: Intra-cluster data grouped by user ID.

ISO 19156, ISO/OGC Observations and Measurements, which defines a conceptual model for observations and for features involved in sampling when making observations. In the context of the SensorThings API, the features are modeled as Things, Sensors and Features of Interest.

7.4.1 Data Harmonization

Each data source provides the data in custom JSON formats, and these are received by the Data Translators, the entry points of information for the back-end. Each translator receives data from a specific source and pushes the data to the Data Quality Control modules as needed. The Data Translator also ensure that the received data respect the format and types of the data models. They use the available information in the received data object, trying to transform it based on the requirements of the SensorThings API standard. In case the data received are not complete or the JSON

Table 7.5: Data evaluation in cluster using the ANOVA table.

Source	**SS**	**df**	**MS**	**F**	**Prob > F**
Group	499	3	166	251	6.47e-34
Error	39	60	0.66		
Total	539	63			

values are not the proper data types, they discard the data object, properly logging this information. In any other case, they forward the transformed data objects to the Data Quality Control module.

For the SensorThings API, the correlation between the received data and the data model involves the creation of two or more data objects for each received measurement. For each data object received, an Observation and a FeatureOfInterest are created containing the measurement collected and the location that is associated with it. In order for these entities to be valid, in the context of the standard, an Observation should be associated with a Datastream. A Datastream is a unique combination of a Thing, a Sensor and an ObservedProperty. Here, the pre-defined the observed properties based on the expected data are presented:

- *Water level;*
- *Water surface velocity;*
- *Air Temperature;*
- *Soil moisture;*
- *LC/LU.*

Each volunteer is modeled as a Thing, based on the unique user identifier that is available in the received metadata. Each portable sensor is modeled as a Sensor, based on the sensor unique identifier included in the metadata. For the data that do not require a portable sensor, the mobile device of the volunteer, uniquely identified by the unique user identifier of the volunteer, is used as a Sensor.

7.4.2 Data Storage

Given the need for a scalable storage solution that can support the provision of spatially dense data with variant time distributions, without a known volume and the need to efficiently respond to spatial and temporal queries from multiple users simultaneously, a distributed storage schema is used for SensorThings API data schema.

The majority of the databases nowadays offer some support for spatial queries. Most of the traditional approaches such as the ones in MySQL [77] and PostGIS [249] are based on R-tree indexes to respond to spatial queries. Such approach has a key challenge, closely related to the spatial distribution of the stored information. R-trees are only effective if they are balanced, meaning that the stored rectangles do not overlap or have empty spaces, as they do not guarantee good worst-case performance. The quality of the R-tree will determine the number of sub-trees that will be examined each time to answer a query. In our case, given that all the measurements are collected at or near PoIs, it is expected that there will be multiple measurements *with the exact same location. The lack of spatial deviation means that it is close to impossible to construct a balanced R-tree.*

Another technique for spatial indexing is the Z-ordering, which aims to alleviate the problems of the R-tree index. The Z-ordering is based on a transformation of the input spatial information to a one-dimensional data structure without any loss of information. While, this approach can handle efficiently spatial points, it lacks the flexibility to efficiently store multiple overlapping objects that lack spatial distribution.

The XZ-ordering [38] is an index that can efficiently handle the lack of spatial distribution. This is achieved by avoiding object duplication and allowing overlapping Z-elements. To this end, in order to ensure the optimal storage and indexing of the data, the XZ-ordering index for the spatial information was chosen for the SensorThings API. This is achieved by using the GeoMesa [141] indexing tools over an Accumulo [122] datastore.

7.4.3 OGC Sensorthings API

We have implemented the SensorThings API as a standalone Tomcat application using the Jersey RESTful Web Services framework. The implementation supports all the requests described in the standard as well as all the filtering capabilities. The implementation has received the compliance badge recognizing that it is an OGC certified product [232]. The endpoints provided are presented below.

GET Requests

Entities may be accessed in many different ways:

- *As a collection of entities: A list of all entities in the specified entity set when there is no service-driven pagination imposed. The response is represented as a JSON object containing a name/value pair named value, a JSON array where each element is a representation of an entity or of an entity reference.*

- *By identifying one entity in a collection: A JSON object of the entity that holds the specified ID in the entity set.*

- *By identifying a property of a single entity: The specified property of an entity that holds the ID in the entity set.*

- *By accessing the value of an entity's property: The raw value of the specified property of an entity that holds the ID in the entity set.*

- *By following a navigation property: A JSON object of one entity or a JSON array of many entities that holds a certain relationship with the specified entity.*

- *By following an association link: A JSON object with a value property, a JSON array containing one element for each associationLink. Each element is a JSON object with a pair, with name URL and the value of selfLinks of the related entities.*

POST Requests

To create an entity in a collection, the client sends a HTTP POST request to that collection's URL. The POST body contains a single valid entity representation. If the target URL for the collection is a navigationLink, the new entity is automatically linked to the entity containing the navigationLink.

Newly created entities may be linked to existing entities, by using the entity ID, or contain the information needed to create the related entities. A request to create an entity that includes related entities, represented using the appropriate inline representation, is referred to as a "deep insert".

PATCH Requests

HTTP PATCH requests are responsible to merge the content in the request payload with the entity's current state, applying the update only to those components specified in the request body. The properties provided in the payload corresponding to properties that can be updated replace the values in the entity. Missing properties of the containing entity or complex properties are not directly altered. These requests may contain binding information for navigation properties. For single-valued navigation properties, this replaces the relationship. For collection-valued navigation properties, this adds to the relationship.

DELETE Requests

A successful DELETE request to an entity's edit URL deletes the entity. The request body must be empty. The implementation implicitly removes relations to and from an entity when deleting it; clients need not delete the relations explicitly. Related entities may be deleted or modified if required by integrity constraints.

7.5 Discussion of Similar Approaches

According to relevant studies [121], the first recorded effort of establishing a Citizen Science project is from 1989, when 225 volunteers across the US were asked to collected rain samples to assist the Audubon Society in an acid-rain awareness campaign. The volunteers collected samples, recorded pH, and reported back to the organization. The information was then used to demonstrate the full extent of the phenomenon [167]. While this project is very important as it established the advantages of engaging volunteers in scientific projects, the data collection process was far from simple; the sample pool was very limited, and the data collected were not validated in any way, making this a poorly designed Citizen Science project by today's criteria.

Since then, many Citizen Science projects have been established, from different scientific areas, aiming to collect contributions from volunteers. To the best of our knowledge, SCENT is the only Citizen Science project tackling the issue of monitoring water parameters such as water level and velocity in the context of aquatic ecosystems, so a direct comparison of the results is not feasible. The design decisions, the system architecture, the validation process as well as the campaign organization will be compared here with other Citizen Science projects that are focusing on monitoring other elements of the environment.

There are many examples of citizen science projects that follow a simple approach to the data collection. They identify the needed information, define simple actions that are to be performed by volunteers, they provide training where needed and they collect the data generated. We present below a few indicative examples.

A very interesting approach was the bumblebee [194] monitoring in the UK that was supported by citizen science. Data were collected on 1022 bumblebee nests when participants were asked to record attributes of bumblebee nests discovered in their gardens and either fill an online form or provide their data with post. The collected data were very valuable, as they made it feasible to study how bumblebees select their habitat, and provided the first quantitative evidence of potential decline in one of the UK's 'big six' common bumblebee species. This was possible due to the wide geographic range in a short time period of the data collection. The main challenge this project faced was the identification of the species. This required photographs that were examined by experts. This is a major scalability issue as an increase in the number of collected data samples would make their validation impractical.

The Neighborhood Nestwatch Program [85] engages citizen scientists in the collection of scientific data and fosters scientific literacy and increased attachment to place in their local natural environment. Nestwatch collects data that can help researchers understand the ecology and population dynamics of eight species of birds along an urban-to-rural gradient in the Washington, D.C., area and teaches people living in urban/suburban settings about bird biology.

eBird [293] is a program launched by the Cornell Lab of Ornithology (CLO) and the National Audubon Society in 2002, which engages a vast network of citizen scientists in reporting bird observations using standardized protocols. The mission is to better understand bird distribution and abundance across large spatiotemporal scales and to identify the factors that influence bird distribution patterns. The collected information provides useful insights on species occurrence, migration timing, and relative abundance at a variety of spatial and temporal scales. To effectively engage birders, eBird provides a permanent repository for their observations and a method for keeping track of each user's personal observations, birding effort and various bird lists. The main challenge of this project is that data are collected by individuals and not by organized campaigns. Inexperienced users might use bird counting techniques wrongly or confuse similar looking bird species. If these users are in locations where not enough information is available, bias may be introduced to the system.

Moon Zoo [267] is proving that there are no limits to where citizen science can go. The project utilises internet crowd-sourcing techniques to take volunteers for a walk around the moon. Moon Zoo users are asked to review high spatial resolution images from the Lunar Reconnaissance Orbiter Camera (LROC), onboard NASAs LRO spacecraft, and perform characterisation such as measuring impact crater sizes and identify morphological features of interest. The tasks are designed to address issues in lunar science and to aid future exploration of the Moon. The information adn measurements derived from the data are also found to align with other estimates for the study area.

In the context of water ecosystems there are many citizen science project. They cover a wide range of research subjects such as the monitoring of species, the pollution and the overall health of the ecosystem. We present below some indicative examples.

CoralWatch [200], launched in 2002, is a citizen-science program that seeks to *integrate education and global reef monitoring by examining coral bleaching and uses a monitoring network to educate the public about reef biology, climate change and environmental stewardship. The main tool for CoralWatch participants is a square of plastic that can be used like a color swatch card, to monitor coral bleaching and therefore coral health. CoralWatch participants match the chart colors to the coral color, record the codes and enter the data on a chart via a website. Scientists developed the colors on the chart using intentionally bleached coral in temperature-controlled aquaria. In this case, the data collection protocol is the main challenge. In order for the information collected to be of use to the scientists, additional data to the color information is needed, putting the user engagement at risk. Special effort was made by the project organizers to communicate with participants through various media.*

Within the context of Coastwatch Europe, an international network of environmental groups, universities and other educational institutions, who in turn work with local groups and individuals around the coast of Europe, many citizen science activities took place aiming to support the monitoring of the coasts around Europe and collect needed information. An annual survey held in UK [162] offered an insight into the major problems and threats to the coastline and raised public awareness regarding environmental issues. Similarly, in Poland [154] the project monitored with the help of volunteers the tendencies in numbers of beverage containers. These containers have posed a problem to the state of popular coastal destinations worldwide for many years and the problem has been addressed in numerous publications that focus on the state of coastlines. The results of the survey indicated that despite the raise in the number of visitors to the coasts the number of containers left behind declined due to the environmental awareness of the public.

The plastic pollution is a fast escalating problem for all the oceans around the world. Careful monitoring and recording of their spatial and temporal distribution, a key requirement to identify the origin and measures to contain the pollution, is a major and expensive task that requires a vast amount of properly distributed data. The citizen science project "National Sampling of Small Plastic Debris" [86] was supported by schoolchildren from all over Chile who documented the distribution and abundance of small plastic debris on Chilean beaches. Thirty-nine schools and nearly 1000 students from continental Chile and Easter Island participated in the activity. To validate the data obtained by the students, all samples were recounted in the laboratory, again raising the issue of scalability with regard to the validity of the data.

In all the Citizen Science projects presented here, the main issue was the data validity. In most cases, expects were tasked with the manual inspection of the collected information or even the duplication of the measurement in order to ensure a high quality of data. Such approaches, however, create the question of the scalability and the reproduction of the experiment in different areas or countries interested in participating. The Scent toolbox, through dedicated tools and applications, has automated the data validation process, creating a scalable solution that can be easily reproduced in other areas of interest.

Taking a different approach to the classical citizen science model where citizens simply collect data for scientist to analyse, the Pathfinder [193] created an online where citizen scientists could not only log the data they collect about sustainability, transportation, and commuting, but also begin to explore it alongside data collected by other citizen scientists, identifying interesting findings, and leveraging these dis-

coveries into collaborative discussions around their data. The results showed that citizen scientists preferred Pathfinder to a standard wiki and were able to go beyond data collection and engage in deeper discussion and analyses.

A very unique approach was adopted by the Zooniverse [280], a platform that hosts many citizen science projects and invites the public to participate in genuine data analysis at a scale that researchers cannot accomplish on their own. The first project to be part of this platform was the Galaxy Zoo [99], launched in July 2007 and successfully engaged 165,000 volunteers in the morphological classification of images of galaxies. Motivated by this successful outcome, the platform expanded to host multiple projects covering a wide range of scientific data for volunteers with different interests. The core methodology is uniform for all the projects, research data is shown to users in the form of images, video and audio via the Zooniverse website. Volunteers are shown how to perform the required analysis via a simple guide or tutorial such that they can then identify, classify, mark, and label them as researchers would do. The results show the success of this approach as Zooniverse citizen science projects have resulted in the classification of more than a million galaxies, the discovery of nearly a hundred exoplanet candidates, the recovery of lost fragments of ancient poetry, and the classification of more than 18,000 thousand wildebeest in images from motion-sensitive cameras in the Serengeti. The combined effort of hundreds of thousands of volunteers adds up to more than 50 years of non-stop effort each year on the Zooniverse platform alone.

Conclusions

An innovative monitoring system and a constellation of technologies that can support the participation of volunteers in the collection of data for environmental ecosystems was presentedin this chapter. Carefully designed tools, easy to use and reliable sensors along with a thorough data quality mechanism allow citizens to contribute proper, unbiased and high quality scientific data as needed for the researchers and local authorities. The collected data are harmonized and offered to environmental scientists and local authorities allowing a more targeted monitoring of areas of interest. The collected information allows scientists to study areas of interest but also is invaluable tool for the policy makers in order to make educated decisions regarding the needs and potential problems of the study area.

Chapter 8

Emotional analysis of short text

The most suitable sources for opinionated text are posts from social media platforms and their monitoring has gained in popularity as more techniques are available for performing sentiment analysis, evaluate the opinions expressed about current events or public figures and characterize them as positive or negative [18]. The term sentiment analysis [189, 5] is used to describe the categorization of text as expressing a positive or negative opinion. Sentiment analysis is performed for entertainment-related events, such as TV series and movies, as well as public figures [92]. In order for such analysis to be successful there is a need for high quantities of annotated text, collected from multiple trustworthy sources and with high time variance.

It is indisputable that this technique has some very important use cases and real-world applications. These use cases, however, are limited to scenarios where the overall opinion about the examined event is not known. There are many scenarios where the positive or negative aspect of the opinion can be assumed with a high degree of confidence. One indicative example is the analysis of text produced by individuals that are exposed to extreme or stressful situations, such as a car accident or an extreme natural disaster. People experiencing such incidents are expected to be very negatively influenced, being worried about their well-being and their properties, angry with the authorities due to lack of preparatory measures or scared about the development of the event and its consequences. Another important challenge during such events is that the opinionated text is produced in a short amount of time, during or close by the occurrence of the event.

Emotion detection [4] in text is proposed as a solution for these challenges. This technique is not focusing on the positive or negative opinions expressed but tries to determine the human emotion that is expressed. The task of identifying the emotions expressed by a person is a very challenging task, that even humans struggle with. Trying to model such identification and create an automated way of identifying the expressed emotion is an even more challenging task, not only due to the limited availability of training datasets but also the restricted information contained in a short text.

Humans more often than not, do not take into consideration the actual words that exchange with other people but base their interpretation regarding the expressed emotion on a series of other cues. The tone used to pronounce the words, the intonation of the phrase, the facial expression of the interlocutor as well as subtle signs from the body language. Even then, it is rare that multiple individuals will have matching opinions when tasked to interpreter the expressed emotion in a situation *that they witness. This is due to the fact that emotion interpretation is affected by personal and social experiences that lead each individual understand and interpret differently emotional expressions.*

A technique to emotionally classify tweets is presented in [34] were tweets are mapped to 6 emotional moods using the Profile of Mood States (POMS) technique. In [181], a text analysis software, the Linguistic Inquiry and Word Count (LIWC), is used to classify tweets into 6 main emotional categories. The LIWC calculates the degree that people use different categories of words across a wide array of texts.The same text analysis software is also used in [323]. All the available solutions so far have two main drawbacks. On the one hand, the emotional categories selected are not based exclusively on the research done by psychologist regarding the emotion theory. Many systems adapt the emotion categories based on their finds, merging or separating emotions, without any consideration to the scientific standard. On the other hand, most of the available systems create their training dataset based on the presence of specific keywords. Unfortunately, using only keywords-based methods, there is no way to validate the accuracy of the classification method as the classifier is almost exclusively trained to recognize these keywords and classify the text accordingly.

A fully-fledged prototype system for the emotion detection in text, as it is published in social media posts has been developed. The system offers a solution for the creation of a fully annotated dataset that can be used for emotion detection and a comparison study between different machine learning models, that were trained using the annotated dataset. This prototype system offers the following:

- *A natural language processor that handles the unique linguistic characteristics of social media posts in regard to lexical, syntax and annotation preferences and provides a uniform text in the annotated dataset.*

- *A hybrid rule-based algorithm that supports the creation of an objectively classified dataset over the Plutchik's eight basic emotions[246]. The algorithm takes into consideration the available emoji in the text and utilized them as objective indicators of the expressed emotion thus efficiently tackling the challenge of the subjectivity of the emotion detection.*

- *An experimental analysis to select the proper machine learning solution, and its proper configuration, for identifying the expressed emotions in text.*

8.1 Methodology

8.1.1 Emotion categories

The discrete emotion theory [261] is one of the most popular theories regarding the emotion categorization, expressed for the first time in 1872 by Charles Darwin [60]. The theory follows the same basic idea behind the color theory, claiming that there are some fundamental emotions that can be used as the base for interpreting and categorizing all the emotions that people may express.

The discrete emotions theory evolved over time and gained in popularity when Paul Ekman presented an extended experimental analysis about the emotions that should be considered as primary and reasons for that [78, 305]. His conclusions *are summarized in two key directions. On the one hand, a pleasant-unpleasant and active-passive scale was identified as capable to depict differences between emotions. On the other hand, emphasis was given to the fact that the emotion interpretation by humans was biased, as the understanding and interpretation of emotions is a skill that people develop though their environment, the other people they communicate with and their social and cultural interactions. Ekman was the first to challenge his own assumptions as too restrictive to map all the human emotions, and tried to establish the proper experiments that would allow the collection of unbiased data.*

Due to his systematic and unbiased study of the emotion classification, Paul Ekman is thought of as a pioneer in the study of emotions and their relation to facial expressions [79]. The results of Ekman's emotion classification study have received a lot of criticism, mainly regarding the way that the data were collected, the process that was chosen for the data validation and the affect these choices had on the integrity of the results. Despite the doubts regarding the process and the results, Ekman has provided the first widely accepted list of the primary emotions, which are happiness, anger, sadness, fear, disgust, and surprise.

Robert Plutchik [246] built upon that and proposed a slightly modified list with eight primary emotions. His classification is based on elements of the psychological evolution of the expression of emotion and it is extracted after careful examination of general emotional responses of individuals for the same event [245]. Plutchik's psycho-evolutionary theory of basic emotions has ten suggestions, which are:

- ***Animals and Humans.*** *Plutchik claims that emotions are also been experienced by animals, especially mammals, and that the concept of emotion is applicable to all evolutionary levels.*

- ***Evolutionary History.*** *According to the psycho-evolutionary theory the emotions appeared through the evolution process, proven to be useful in certain situations and have evolved into various forms of expression in different species.*

- ***Survival Issues.*** *A core idea of evolution is the natural selection and the idea that physiological changes are the result of the need of a species to survive. Similarly, emotions evolved overtime and helped with key survival issues posed by the environment. All the eight emotions can be mapped to survival needs, for example anger helps humans fight predators, anticipation forces people to prepare and surprise to learn.*

- ***Prototype Patterns.*** *As with the primitive color theory, similarly for the emotions, it is believed that there are certain common emotional elements that can be used to identify prototype emotions, despite the different forms of expression in different species.*

- ***Basic Emotions.*** *There are eight primary emotions, joy, trust, fear, surprise, sadness, disgust, anger and anticipation.*

- ***Combinations.*** *Any other emotions experienced by humans, even though they have their own names, can be traced back to various combinations, mixtures and intensities of the eight primitive emotions.*

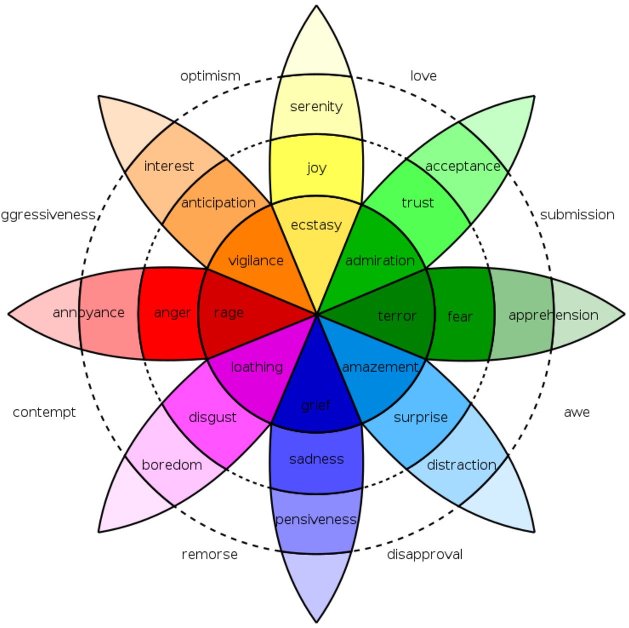


Figure 8.1: Plutchik's wheel of emotions. As provided by Machine Elf 1735 - Own work, Public Domain

- ***Hypothetical Constructs.*** *Primary emotions are recognized as ideas created to describe experiences and are mainly idealized states with specific properties and characteristics.*

- ***Opposites.*** *The eight emotions are designed as pairs of polar opposites. The pairs are (joy, sadness), (fear, anger), (anticipation, surprise) and (disgust, trust).*

- ***Similarity.*** *The human emotions vary in the degree of similarity to one other. There are pairs of emotions such as (aversion, disgust) that are referring to very similar emotional states while others cannot be compared, such as (admiration, fury).*

- ***Intensity.*** *Plutchik suggested that emotions can be expressed in varying degrees of intensity.*

The Plutchik's eight average-intensity emotional categories, which are joy, trust, fear, surprise, sadness, disgust, anger and anticipation, will be used as the classes for the emotion detection system proposed here. The system will offer an observation percentage for each emotion allowing the extraction of more complex emotions as it is defined by Plutchik's wheel of emotions, as shown in Figure 8.1.

8.1.2 Dataset Acquisition

Computers are unable to understand the human language without any interpretation, a functionality that is considered useful for many applications including automated assistants and smart networks. The task of machines to understand the human language is very challenging as it is very complicated, it includes expression of emotion

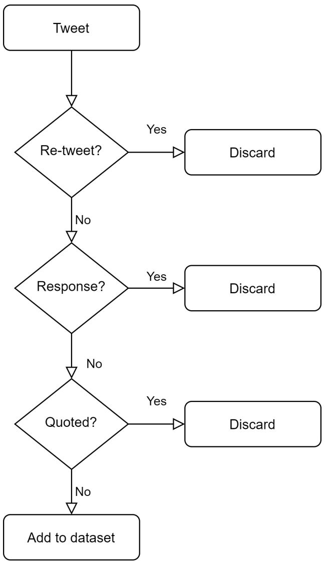


Figure 8.2: Tweet collection process.

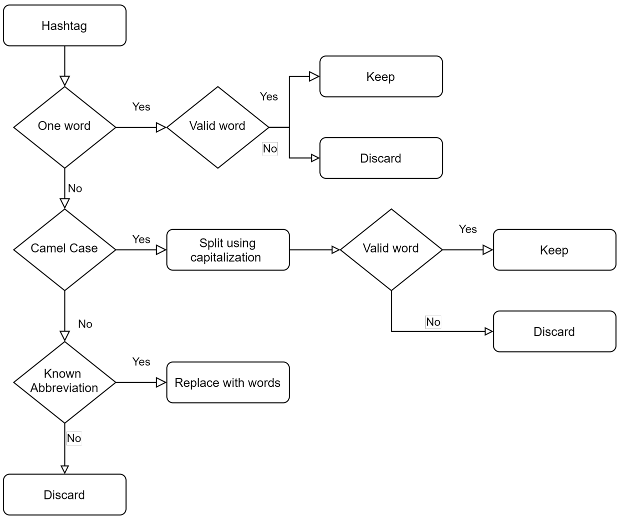


Figure 8.3: Hashtag processing in tweets

and usage of lax syntactical and grammatical rules. Aiming to alleviate this barrier, there has been an extended research regarding the ways that artificial intelligence solutions can help computers with the interpretation of the human language as well as support machines in communicating in a human-like way[68].

This research field, called Natural Language Processing (NLP) [157], has as a key target the development of automated solutions that will be able to receive some text and process it in a way that will enable its interpretation given the overall context. Unofficially, the aim of the NLP can be defied as the process that will allow machines to understand a text, the same way that an individual would.

The main challenge for the NLP solutions is the complexity of the meaning extraction. Understanding the individual meaning of the words when isolated is trivial but identifying the meaning when they are in phrases, accompanied with irony or sarcasm, is very challenging. Irregular verbs, plural number exceptions, double negatives used either to emphasize a point or to create a positive as well as multiple meanings of words based on the overall context are some of the most interesting peculiarities of the human language [198].

Humans tend to interpret a word based on the context and the overall meaning very easily, as it is something that they are accustomed to. Modeling such behavior, however, is very challenging as this is not a deterministic task. A very indicative example is the word bark which can be used either as a verb or a noun with a plethora of meanings. The following phrases, are all valid uses of the term but its interpretation is completely different. 'The dogs bark every time there is a strange at the yard', 'Some kids wrote their names in the tree bark', 'I love pretzels covered with almond bark', 'Do you know if bark beetle causes problems with the development of the trees?'.

The task of performing NLP analysis to social media posts comes with some additional challenges. It is important to note that the most the well-established solutions have been trained using text coming from journal articles and encyclopedias [91], sources of large quantities of high quality text [33], that comply with spelling and syntax rules, use proper grammar and vocabulary that can be found in official *dictionaries. This is a complete contrast with the characteristics of the language used at social media posts. In detail, the identified differences in the language are:*

- ***Text size.*** *Traditional NLP systems have been trained using large passages of text that have only one author. Social media posts have limited characters and are produced by multiple individuals.*

- ***Topic diversity.*** *Traditional sources provide content that presents specific topics within the same text, journal articles for example are expected to be focused on one single topic throughout. A single social media post can included multiple topics, and express opinions of different intensity, at the same time that multiple users are discussing a plethora of topics that are of interest.*

- ***Vocabulary & Spellcheck.*** *Formal text includes exclusively words that can be found in dictionaries, used following the proper spelling and the word capitalization rules. In social media posts, users often use words that are non-existing, either formed on the fly to emphasize a situation or due to incorrect spelling. There are also some habits of these users that deviate from the proper spelling rules, such as the usage of letter repetition to give emphasis, the use of capital letters to expressed the intensity of their emotions or the shortening of words to their sound for shorter messages, like 'heyyyyy', 'I am SAD' or 'u'.*

- ***Syntax & Grammar.*** *Formal text follows all the syntax and grammar rules, including the use of punctuation marks. In social media posts, emphasis is prioritized over proper usage of the linguistic rules. The text includes incomplete phrases, non-existent grammar, phrases without verbs or subjects, incorrect and excessive use of punctuation marks.*

- ***Text objectivity & originality.*** *Formal sources are providing unique and objective text that contains properly presented facts and precise information with limited emotional expressions. On the other hand, social media posts are often repetitions or additions to already presented opinions. The posts aim to present specific views and arguments over recent events and topics of interest, often expressed under emotional excitement.*

- ***Abbreviations.*** *Shortened text is really used in official documents, unless it has been clearly defined, is related to a well-established term that is used throughout the document and is properly explained. The users of the social media are creating abbreviations continuously, giving them multiple interpretations based on the overall context and fail to explain their meaning.*

Dataset Collection & Harmonization

Twitter is the social media platform selected for the collection of the text that will be used as it is focused mainly on text messages and not on video and images. The tweets will be pre-processed in order to be harmonised and annotated using a rule based approach[46]. The approach will ensure that the special characteristics of the social media posts are taken into account, and utilized as needed in order to objectively annotate the collected posts, and provide a high quality training dataset.

A dedicated developer's account was created for the Twitter platform[310] to obtain the proper permissions and authentication criteria in order to collect social media posts. The Twitter Streaming API[311] was used through the Python Tweepy[260] library. The library is handling the proper communication as well as the management of respecting the post capturing quotas of the Twitter API, allowing us to focus on developing the logic needed for the proper data collection.

The main feature of the Twitter Streaming API is the need to provide a filter to be used for the streaming. The filter includes multiple criteria, including keywords, location and language [95] that can be used independently. Many different configurations were considered regarding the best way to collect the training dataset, including using directly as keywords the emotions and their synonyms or capturing only tweets that contained emoji. These configurations however were quickly dismissed, fearing that such dataset will lack the needed heterogeneity to categorize with accuracy any given text.

After investigating the most popular, based on their frequency of use, words in tweets[300], a list was composed and used as the streaming filter. The list is {"a", "the", "I", "to", "you", "in", "on", "for", "with", "that"}. In addition, understanding that the profiles of the users, and inevitable the content of the posts, changes based on the day and time they are produced, for example teenagers tend to use social media late at night while parents of young children in the afternoon, special consideration was given to the collection of posts at different times and days.

As an additional step to ensure the collection of a high quality dataset, retweets, quotes and responses to tweets were not included in order to avoid text repetition and phrases too small to be interpreted on their own. The process of eliminating such tweets is shown in Figure 8.2.Having established the methodology to collect the social media posts, the next step is to harmonize their content, focusing on the following:

Hashtags: *Hashtags are a unique characteristic of tweets, they are used either at the beginning of important to the post's content words, providing a indirect categorization or at the end of the post, along with words and short phrases that have special meaning and importance to the post. For example, a user at the airport ready to leave for summer vacations would write 'Waiting at the #airport, the flight is leaving for my #vacations in less than an hour!!! #Summer #SummerTime #traveling'. Such tweet is using the hashtags to highlight important words for the post as well as to provide additional context.*

A innovative data flow has been designed, as presented in Figure 8.3, to examine each hash-tagged word and identify the words included. The first step is to remover the hashtag and look up the word in a dictionary, if the word is found there then the hash-tagged word is replaced by it, for example #vacations is replaced by vacations. Next, it is examined if the hash-tagged word is actually multiple words written in camel case. If this is the case, then the capitalization rule is used to split the hash-tagged word into multiple ones. Again, each split word is validated through a dictionary to ensure that the division was accurate, for example #SummerTime is replaced by 'summer time'. As dictionary, the nltk text corpora and lexical resources[228] are used. Last but not least, the hash-tagged word is checked over the Abbreviations[2] open API and replaced with the corresponding words in case of a match. If none of the above steps are fruitful then the hash-tagged word is considered a misspell and is removed from the tweet.

URLs: *Tweets are known for containing hyperlinks to other sources, as they* *add value and meaning to the posts. Such links however are not of any value for the emotion detector and their unusual spelling is more probable than not to confuse the feature extractor. For this reason, they are removed from the tweets using the Tweet pre-processor Python library[309].*

Mentions: *As discussed also for the hyperlinks, mentions are very important for the Twitter social media platform. They are used mainly to responses and when specific people are referenced. They always start with the special character @ followed with the user name of the person mentioned. While their purpose is important for the interaction at the platform, they have no semantic value. On the contrary, their unique spelling and the sparsity of their appearance in the dataset may confuse the feature extractor and the emotion detector. For this, it has been decided to remove them from the text included in the dataset.*

Character repetition/ misspelled words: *Each tweet is divided into words, and each word is check against the dictionary to determine its validity. In case of an invalid word, the phenomenon of character repetition is investigated, gradually eliminating characters that appear more than once and checking again if the word is valid. Posts with misspelled or invalid words are removed from the dataset.*

Emoji: *The Unicode codes of the emoji are not processed at this time in any way. The emoji are very important, as it will be discussed in detail in Section 8.1.2, for the rule-based classification of the tweets in the eight categories. For this reason, they are not modified in the harmonization phase.*

Dataset Annotation

Having collected and harmonized the dataset, the next step is to provide proper categorization in the eight emotion categories of the Plutchik's wheel. This is achieved through a rule based python script, that complies with the following rules:

Emoji: *The emoji that are used in excess in social media posts, as collected by the emoji python library[83], were examined in detail and separated into eight categories corresponding to the eight emotion categories of the Plutchik's wheel. A large percentage of the emoji was not included in any of the eight categories, as they were general and descriptive, not associated with any emotional state. Indicative examples of such emoji are vehicles, the fruits and vegetables as well as objects and animals. Another important part of the emoji dataset that was not categorized, included emoji that were referring to emotions that were a combination of more than one of the eight emotion categories according the Plutchik's wheel, such as remorse and aggressiveness.*

Approximately, only 7% of the initial emoji were categorized as expressing one of the eight emotions. The number of emoji per category differs a lot, with categories such as anger and joy having up to 60 emoji while categories such as anticipation and disgust having less than 30. Aiming to ensure that the emoji included in the lists for each emotion are popular, and consequently probable to be found in posted tweets, the most commonly used emoji as published in tweets and monitored by a live web tool, the Emoji Tracker [84], were also examined. The lists with the emoji were updated to include some of the emoji found in the list.

For each collected tweet, the emoji that it contains are examined against the eight lists. If the emoji is not present in any of the lists then it is simply replaced by its corresponding text. In the case that only one of the emoji of the tweet is in one of the lists then the tweet is annotated as belonging to that category, also if more than *one emoji belongs to the same list, again the tweet is annotated as belonging to that category. In both cases the emoji that were used for the categorization of the tweet are removed from the post, and not replaced by their text. If multiple emoji belong to multiple lists then there is no way to properly annotate the tweet, so the emoji are replaced by their corresponding text [177].*

NRC Emotion Lexicon: *The NRC Emotion Lexicon[212] is a list of English words and their associations with eight basic emotions of the Plutchik's wheel. Each tweet is examined in case such word is present, and if so classified accordingly.*

Lexical relations: *WordNet [93] is a large lexical database of English, where nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms, so that each set is expressing a distinct concept. The sets of synonyms are also interlinked by means of semantic and lexical relations. These relationships are synonymy (car,automobile), antonymy (hot, cold), hyponym (red, colour), hypernymy (cutlery, fork) and meronymy (tree, forest).*

In the annotation rules, two groups of relationships are identified. The synonyms/hyponyms/hypernyms that are providing for each emotion a set of words and the antonyms of each emotion that are added in the set of words for the polar opposite of the emotion based on the Plutchik's wheel [175, 176]. The annotation process for the tweets is shown in Figure 8.4. Taking into consideration that some emotions are expressed more frequently, the final dataset has a different number of tweets per emotion category. Tests performed in different hours during the day and during different days of the week, including late at night and during weekends showed that there is a constant lack of social media posts expressing fear and anticipation. Tweets expressing anger and trust are also hard to collect, while joy and surprised are the most common emotions expressed. The collected dataset can be used to train any emotion detection machine learning model, following the process depicted in Figure 8.5.

8.1.3 Classification model development

Amongst many classification methods, the Long short-term memory networks (LSTM) are wide used to classify text and social media posts. LSTM [138] is a type of recurrent neural network (RNN) [82], that specifically addresses the issue of learning long-term dependencies. Their architecture allows for the accumulation of information during operation and uses feedback to remember previous network call states [229]. A generic description of the network model used in this document is presented in Figure 8.6 where each circle represents an LSTM cell, the input is given as a vector representing a tweet and Output h will return the result that is of interest. Detailed description of the architecture of LSTMs can be found in [235].

LSTM networks are often used in time-series forecasting and pattern analysis, such as on petroleum production [264], on weather [160] and on fog data [206]. Also, in the literature the Long-Short-Term-Memory (LSTM) networks are the leading methodology for text analysis and classification. In [229] the LSTM is used to classify pre-labeled texts gathered from online forums in the categories of spam and not-spam; also the content of books reviews are classified into positive or negative. In [251] three public available pre-classified datasets, a movie review dataset and two sets with restaurant reviews, are used in order to train an LSTM to classify the reviews into negative and positive ones. Also, in [322] pre-labeled articles as news

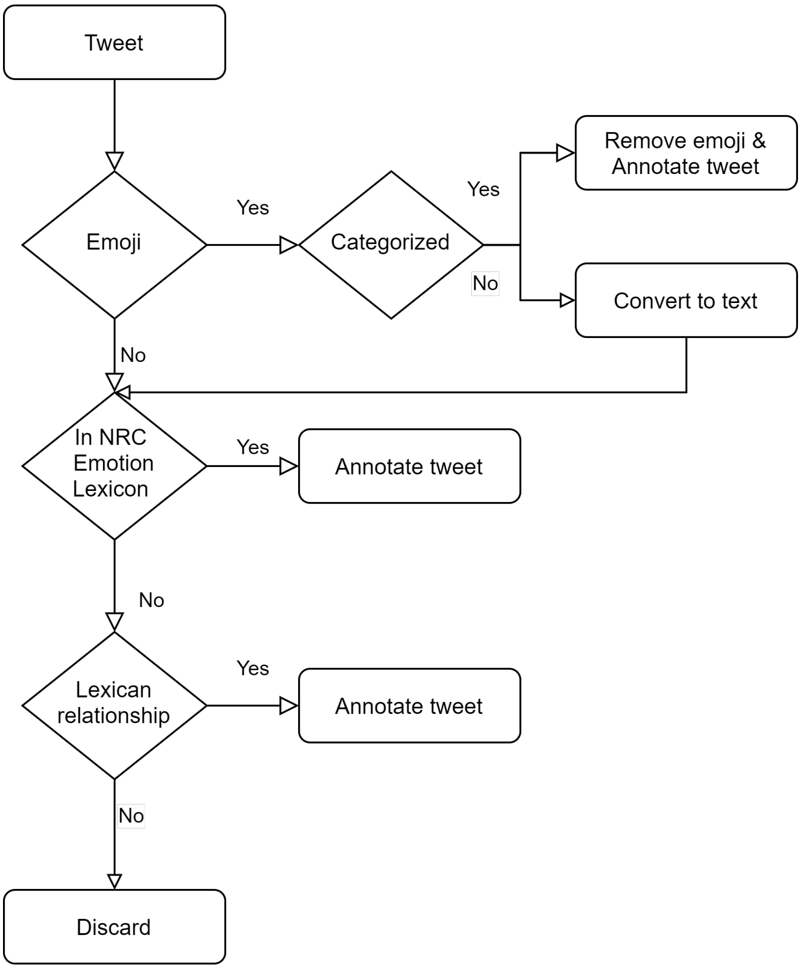


Figure 8.4: Rule based tweet annotation

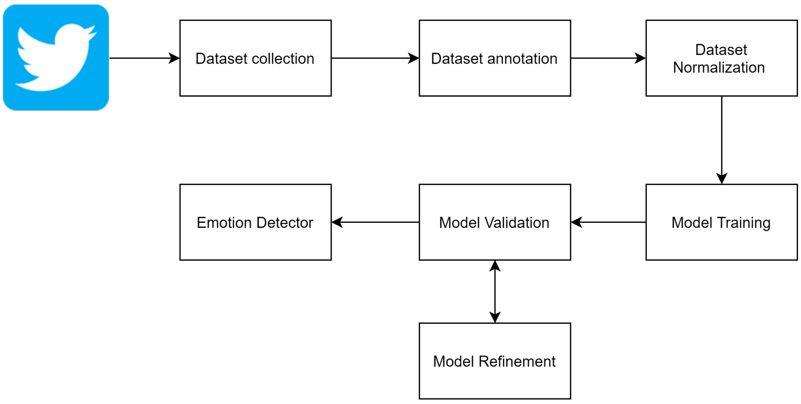


Figure 8.5: Dataset usage in model training

and reviews of movies and products are classified in the categories of positive and negative. Finally, in [250] posts from social media referring to political beliefs are classified as Democratic or Republican.

For the purposes of this paper, an LSTM network is trained to solve the presented classification problem of the emotion detection. In order to compare the results of the LSTM classifier with other methodologies, five other classifiers have been trained using the same dataset. These baseline classifiers are:

- *A linear support vector machine using the stochastic gradient descent classifier*

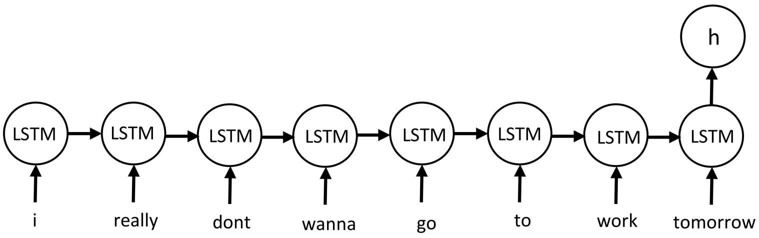
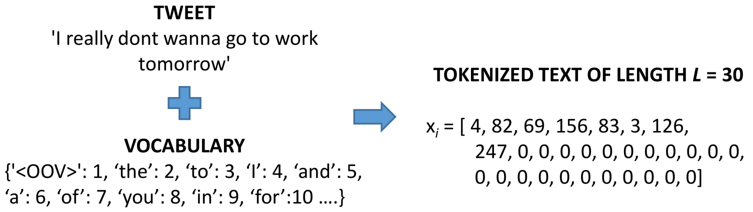


Figure 8.6: The sequential learning procedure of an LSTM network.


TWEET
'I really dont wanna go to work tomorrow'
VOCABULARY
TOKENIZED TEXT OF LENGTH *L* = 30

{'<OOV>': 1, ‘the’: 2, ‘to’: 3, ‘I’: 4, ‘and’: 5, ‘a’: 6, ‘of’: 7, ‘you’: 8, ‘in’: 9, ‘for’:10 ....}

x_i = [4, 82, 69, 156, 83, 3, 126, 247, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Figure 8.7: The tokenization process of the tweets. OOV refers to ‘out of vocabulary’. As the reader can see from this figure a tweet may not always be coherent.

(called SVM-SGD in this document) [36]

- *An XGBoost classifier [50]*
- *A Naive Bayes classifier for multinomial models [168]*
- *A Decision Tree classifier[342]*
- *A random forest classifier [6]*

In order to train all the above methods, the collected posts of Twitter had to be transformed into numbers. Using the downloaded tweets, a vocabulary was created using the V most common words appearing. The parameter V also determines the dimension of the feature space used for the classification problem. A higher dimension could presumably capture more information and obtain better results, at the cost of slower training times. Using the created vocabulary, each tweet was transformed to a vector of length L representing the words it contains; this process is called tokenization. If a tweet contains less that L words, then it is filled with zeros. Figure 8.7 presents this procedure.

To perform the training process of the classifiers, the average loss of cross-entropy, also known as Negative Log Likelihood Loss, objective function was minimized:

$$J(\mathbf{w}) = -\frac{1}{N}\sum_{i=1}^{N}[y_i\text{log}(\hat{y}_i) + (1-y_i)\text{log}(1-\hat{y}_i)] \tag{8.1}$$

where $\boldsymbol{w}$ represents all the synaptic weights used in the LSTM, N is the size of the dataset, y_i and $\hat{y}_i$ symbolize the number of the original and predicted category (y_i and $\hat{y}_i \in \{1, 2, ..., 8\}$ in this document). As equation (8.1) gets minimized, more of the data are being classified into the correct category. For this optimization problem, the Adam algorithm was used [170].

For all the simulations the Python 3.6.10 programming language was used with the libraries xgboost 1.2.0 [49], scikit-learn 0.23.2 [242] and tensorflow 2.4.1 [201].

Classifier	Testing accuracy
LSTM	**91.90%**
SVM-SGD	86.86%
XGBoost	84.45%
Naive Bayes	77.01%
Decision Tree	84.69%
Random forest	80.35%

Table 8.1: Performance of the classification methods applied on the testing dataset

#	**Correctly Classified Tweets**	**Annotated as**	**Classified as**
(1)	Nothing hurt more than loyalty coming from one side in a relationship.	anger	anger
(2)	What a vibe This song made my day	joy	joy
(3)	How you gonna find a knockoff version of me that welcomes commitment that's gross	disgust	disgust
(4)	The greatest day of the year has finally begun	anticipation	anticipation
(5)	How to cope with parents who regret your existence	sadness	sadness
(6)	can we cancel 2020? im so done	sadness	sadness
(7)	god bless the ability to mute people on instagram	anticipation	anticipation
(8)	I cant wait to live alone in the mountains.	anticipation	anticipation
(9)	I hate this fucking song	anger	anger
	Incorrectly Classified Tweets		
(10)	40 minutes till i can play tomb raider again	sadness	joy
(11)	Man I hate my life	disgust	anger
(12)	allergy highs	disgust	joy
(13)	CHANGE MY PIC TO ALL THOSE DIRTY HACKERS AND SCAMMERS TRYING TO USE MY FB, INSTAGRAM PG, THANK YOU MONIE FOR THE LOOKOUT CHICK!!! #CLASS OF '88	joy	anger
(14)	I love not being loved by my friends	joy	sadness

Table 8.2: Examples of the classified tweets

8.2 Results

A total of 1.2 million annotated tweets was downloaded using the process described in Section 8.1.2;this number of tweets is just a balanced (among the emotion categories) subset of a greater set of 3.6 tweets that were collected with a speed of about 700,000 tweets per day. This dataset is publicly available at `https://www.kaggle.com/tasos123/tweets-annotated-to-emotions` and it contains only the tweet-ID and the class it has been annotated in order to protect the anonymity of the Tweeter's

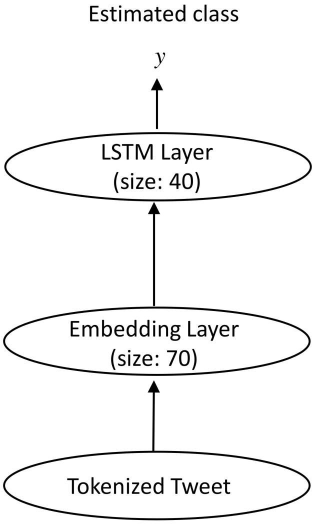


Figure 8.8: The layers used in the training process of the LSTM network.

users. The 80% of them were used as a training sample, 10% as validation and 10% as testing. The used vocabulary size is $V = 20{,}000$ and each tweet was transformed to a vector of length $L = 50$. Choosing other values for the parameters V and L was leading to similar or worse results. Also greater values for V were resulting to very slow training procedures or were resulting to overfitting networks and were avoided. It should be noted that all computations were performed to a CPU (an Intel i7-4770 @ 3.40GHz) and not any GPU was used in all the experiments of this document.

Fig. 8.8 shows the layers used in the training process of the LSTM together with their parameters. For the baseline classifiers, the default parameters (according to their libraries mentioned in the previous section) were used.

In order to compare the performance of the classification methods, the overall accuracy on the testing dataset is used. Table 8.1 shows the results of the LSTM compared with the five other baseline classifiers. The LSTM provides the best overall performance and the SVM-SGD follows. The Naive Bayes model provides the worse performance than all the others, but this could be resulted by the use of the partial fitting function that was used since it was demanding huge amounts of RAM memory. Figure 8.9 shows the LSTM's accuracy in each of the eight classes at the form of a confusion matrix. This matrix was normalized, i.e. each row sums to one. Higher values in the main diagonal of this matrix show better performance for each category.

Disgust and joy are the emotions that are better predicted using the LSTM, while trust and anticipation are the emotions with the lowest network performance. The differences, however, in the performance of the LSTM in the eight categories are such that can be considered insignificant. Especially for trust and anticipation, the two least accurate predicted emotions, careful examination of the confusion matrix shows that text expressing trust is more often than not wrongly classified as anticipation and vice versa. Based on this observation it can be assumed that, even though not anticipated, there is some overlap in phrases and expressions used to express these emotions. It is also worth noticing that these emotions are the ones that have the least keywords associated with them, the fewest synonyms and the least diversity regarding emoji expressing them. For these reasons the network's accuracy is limited in these categories.

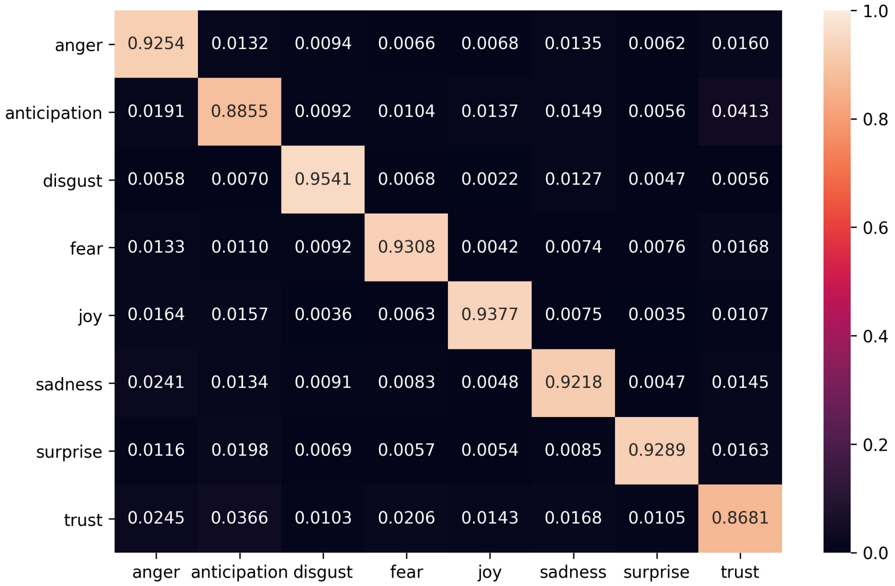


Figure 8.9: The Confusion matrix with the classification results of the LSTM network

Table 8.2 presents some examples of classified tweets, nine correctly classified ones and five classified to the wrong class. It can be observed that, the second wrongly classified tweet, (#11 in Table 8.2), although it was labeled as disgust in the annotated dataset due to the presence of an emoji, the LSTM classified it to the anger category, possibly using the "I hate" sequence of words. Similar behaviour is presented in the tweet #13 that is was labeled as joy in the annotated dataset due to the presence of a smiling emoji, but from the human perspective it is more likely that it belongs in the anger class, like the LSTM classified it. Also in #14 we can see the presence of sarcasm/irony that although it was labeled in the annotated dataset as joy due to the existence of the "love" keyword, it was classified by the LSTM to the more fitting sadness emotion.

Bibliography

[1] *A Structural View of Biology*. https://www.rcsb.org/. 2018.

[2] *Abbreviations*. https://www.abbreviations.com/. Accessed: 2020/11/17.

[3] James Abello, Frank van Ham, and Neeraj Krishnan. "ASK-GraphView: A Large Scale Graph Visualization System". In: *TVCG* 12.5 (2006).

[4] Francisca Adoma Acheampong, Chen Wenyu, and Henry Nunoo-Mensah. "Text-based emotion detection: Advances, challenges, and opportunities". In: *Engineering Reports* (2020), e12189.

[5] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca J Passonneau. "Sentiment analysis of twitter data". In: *Proceedings of the workshop on language in social media (LSM 2011)*. 2011, pp. 30–38.

[6] Yassine Al Amrani, Mohamed Lazaar, and Kamal Eddine El Kadiri. "Random forest and support vector machine based hybrid approach to sentiment analysis". In: *Procedia Computer Science* 127 (2018), pp. 511–520.

[7] Fahad Alahmari, James A. Thom, Liam Magee, and Wilson Wong. "Evaluating Semantic Browsers for Consuming Linked Data". In: *ADC*. 2012.

[8] *Allie Abbreviation And Long Form Database in Life*. http://data.allie.dbcls.jp/sparql. 2019.

[9] *Alpine Ski Racers of Austria*. http://vocabulary.semantic-web.at/PoolParty/sparql/AustrianSkiTeam. 2019.

[10] Oszkár Ambrus, Knud Möller, Siegfried Handschuh, et al. "Konduit VQB: a visual query builder for SPARQL on the social semantic desktop". In: *Workshop on visual interfaces to the social and semantic web*. 2010.

[11] J Chris Anderson, Jan Lehnardt, and Noah Slater. *CouchDB: the definitive guide: time to relax*. " O'Reilly Media, Inc.", 2010.

[12] Renzo Angles and Claudio Gutierrez. "Survey of graph database models". In: *ACM Computing Surveys (CSUR)* 40.1 (2008), pp. 1–39.

[13] Daniel Archambault, Tamara Munzner, and David Auber. "Grouse: Feature-Based, Steerable Graph Hierarchy Exploration". In: *EuroVis*. 2007.

[14] Kevin Ashton et al. "That 'internet of things' thing". In: *RFID journal* 22.7 (2009), pp. 97–114.

[15] Ghislain Auguste Atemezing and Raphaël Troncy. "Towards a linked-data based visualization wizard". In: *COLD*. 2014.

[16] Luigi Atzori, Antonio Iera, and Giacomo Morabito. “The Internet of Things: A survey”. In: *Computer Networks* 54.15 (2010), pp. 2787–2805. ISSN: 1389-1286. DOI: `https://doi.org/10.1016/j.comnet.2010.05.010`. URL: `http://www.sciencedirect.com/science/article/pii/S1389128610001568`.

[17] David Auber. “Tulip - A Huge Graph Visualization Framework”. In: *Graph Drawing Software*. 2004.

[18] Rushlene Kaur Bakshi, Navneet Kaur, Ravneet Kaur, and Gurpreet Kaur. “Opinion mining and sentiment analysis”. In: *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE. 2016, pp. 452–455.

[19] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. “Gephi: An Open Source Software for Exploring and Manipulating Networks”. In: *ICWSM*. 2009.

[20] Alexander De Leon Battista, Natalia Villanueva-Rosales, Myroslav Palenychka, and Michel Dumontier. “SMART: A Web-Based, Ontology-Driven, Semantic Web Query Answering Application.” In: *Semantic Web Challenge* 295 (2007).

[21] Rudolf Bayer and Edward McCreight. “Organization and maintenance of large ordered indexes”. In: *Software pioneers*. Springer, 2002, pp. 245–262.

[22] Satanjeev bbanerjee and Ted Pedersen. “Extended gloss overlaps as a measure of semantic relatedness”. In: *Ijcai*. Vol. 3. 2003, pp. 805–810.

[23] *BBC John Peel sessions from DBTune*. `http://dbtune.org/bbc/peel/cliopatria/yasgui/index.html`. 2019.

[24] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. “The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles”. In: *SIGMOD Rec.* 19.2 (May 1990), pp. 322–331. ISSN: 0163-5808. DOI: `10.1145/93605.98741`. URL: `https://doi.org/10.1145/93605.98741`.

[25] Jon Louis Bentley. “Multidimensional binary search trees used for associative searching”. In: *Communications of the ACM* 18.9 (1975), pp. 509–517.

[26] Jon Louis Bentley. “Multidimensional divide-and-conquer”. In: *Communications of the ACM* 23.4 (1980), pp. 214–229.

[27] Stefan Berchtold, D. Keim, and H. Kriegel. “The X-tree : An Index Structure for High-Dimensional Data”. In: *VLDB*. 1996.

[28] Tim Berners-Lee, Yuhsin Chen, Lydia Chilton, Dan Connolly, Ruth Dhanaraj, James Hollenbach, Adam Lerer, and David Sheets. “Tabulator: Exploring and analyzing linked data on the semantic web”. In: *ISWUIW*. Citeseer. 2006.

[29] Tim Berners-Lee and Mark Fischetti. *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*. DIANE Publishing Company, 2001.

[30] Adrian Bielefeldt, Julius Gonsior, and Markus Krötzsch. “Practical Linked Data Access via SPARQL: The Case of Wikidata.” In: *LDOW WWW*. 2018.

[31] Nikos Bikakis, Melina Skourla, and George Papastefanatos. “rdf:SynopsViz - A Framework for Hierarchical Linked Data Visual Exploration and Analysis”. In: *ESWC*. 2014.

[32] *Bio2RDF*. `https://bio2rdf.org/sparql`. 2019.

[33] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O’Reilly Media, Inc.", 2009.

[34] Johan Bollen, Alberto Pepe, and Huina Mao. *Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena.* 2009. arXiv: `0911.1583 [cs.CY]`.

[35] Mathias Bonduel, Mads Holten Rasmussen, Pieter Pauwels, Maarten Vergauwen, and Ralf Klein. “SPARQL-visualizer: A Communication Tool for Collaborative Ontology Engineering Processes”. In: ().

[36] Léon Bottou. “Stochastic gradient descent tricks”. In: *Neural networks: Tricks of the trade*. Springer, 2012, pp. 421–436.

[37] Judith S Bowman, Sandra L Emerson, and Marcy Darnovsky. *The practical SQL handbook: using structured query language*. Addison-Wesley Longman Publishing Co., Inc., 1996.

[38] Christian BÖxhm, Gerald Klump, and Hans-Peter Kriegel. “XZ-Ordering: A Space-Filling Curve for Objects with Spatial Extension”. In: *Advances in Spatial Databases*. Ed. by Ralf Hartmut Güting, Dimitris Papadias, and Fred Lochovsky. 1999.

[39] Christian BÖxhm, Gerald Klump, and Hans-Peter Kriegel. “Xz-ordering: A space-filling curve for objects with spatial extension”. In: *International Symposium on Spatial Databases*. Springer. 1999, pp. 75–90.

[40] Ulrik Brandes. “On variants of shortest-path betweenness centrality and their generic computation”. In: *Social Networks* 30.2 (2008), pp. 136–145.

[41] Josep Maria Brunetti, Sören Auer, and Roberto García. “The Linked Data Visualization Model.” In: *ISWC*. 2012.

[42] Josep Maria Brunetti, Sören Auer, Roberto Garcıa, Jakub Klımek, and Martin Nečaský. “Formal linked data visualization model”. In: *Proceedings of ICIIWAS*. ACM. 2013.

[43] J Buckley et al. “The internet of things: from RFID to the next-generation pervasive networked systems”. In: *Auerbach Publications, New York* (2006).

[44] Howard Butler, Martin Daly, Allan Doyle, Sean Gillies, Stefan Hagen, Tim Schaub, et al. “The geojson format”. In: *Internet Engineering Task Force (IETF)* (2016).

[45] Diego Valerio Camarda, Silvia Mazzini, and Alessandro Antonuccio. “LodLive, exploring the web of data”. In: *Proceedings of the 8th International Conference on Semantic Systems*. 2012, pp. 197–200.

[46] Lea Canales and Patricio Martinez-Barco. “Emotion detection from text: A survey”. In: *Proceedings of the Workshop on Natural Language Processing in the 5th Information Systems Research Working Days (JISIC)*. 2014, pp. 37–43.

[47] Josiah L Carlson. *Redis in action.* Manning Publications Co., 2013.

[48] Šejla Čebirić, François Goasdoué, Haridimos Kondylakis, Dimitris Kotzinos, Ioana Manolescu, Georgia Troullinou, and Mussab Zneika. "Summarizing semantic graphs: a survey". In: *The VLDB Journal* (2019).

[49] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* KDD '16. San Francisco, California, USA: ACM, 2016, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: `10.1145/2939672.2939785`. URL: `http://doi.acm.org/10.1145/2939672.2939785`.

[50] Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining.* 2016, pp. 785–794.

[51] Yu-Kai Chou. *Actionable Gamification: beyond points, badges, and leaderboards.* Octalysis Media, 2015.

[52] Edgar F Codd. "A relational model of data for large shared data banks". In: *Software pioneers.* Springer, 2002, pp. 263–294.

[53] Douglas Comer. "Ubiquitous B-Tree". In: *ACM Comput. Surv.* 11.2 (June 1979), pp. 121–137. ISSN: 0360-0300. DOI: `10.1145/356770.356776`. URL: `https://doi.org/10.1145/356770.356776`.

[54] Cathy C. Conrad and Krista G. Hilchey. "A review of citizen science and community-based environmental monitoring: issues and opportunities". In: *Environmental Monitoring and Assessment* 176.1 (May 2011), pp. 273–291. ISSN: 1573-2959. DOI: `10.1007/s10661-010-1582-5`.

[55] Copernicus. *Copernicus Land Monitoring Service.* `https://land.copernicus.eu/`. 2019.

[56] Copernicus. *CORINE Land Cover nomenclature conversion to Land Cover Classification system.* `https://land.copernicus.eu/eagle/files/eagle-related-projects/pt\_clc-conversion-to-fao-lccs3\_dec2010`. 2010.

[57] Courtney Corley and Rada Mihalcea. "Measuring the semantic similarity of texts". In: *Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment.* Association for Computational Linguistics. 2005, pp. 13–18.

[58] CouchDB. `https://couchdb.apache.org/`. 2020.

[59] Aba-Sah Dadzie and Matthew Rowe. "Approaches to visualising Linked Data: A survey". In: *Semantic Web* 2.2 (2011).

[60] Charles Darwin and Phillip Prodger. *The expression of the emotions in man and animals.* Oxford University Press, USA, 1998.

[61] Chris Date et al. *Database in depth: relational theory for practitioners.* " O'Reilly Media, Inc.", 2005.

[62] *Datos.* `http://datos.bcn.cl/sparql`. 2019.

[63] *DBpedia.* `http://dbpedia.org/sparql/`. 2019.

[64] DDNI. *Danube Delta National Institute for R&D.* `http://ddni.ro/wps/`. 2020.

[65] Samur FC De Araujo and Daniel Schwabe. "Explorator: a tool for exploring RDF data through direct manipulation". In: *DOW2009*. 2009.

[66] Andrea De Mauro, Marco Greco, and Michele Grimaldi. "A formal definition of Big Data based on its essential features". In: *Library Review* 65.3 (2016), pp. 122–135.

[67] Andrea De Mauro, Marco Greco, and Michele Grimaldi. "What is big data? A consensual definition and a review of key research topics". In: *AIP conference proceedings*. 2015, pp. 97–104.

[68] *Deep Learning for NLP: An Overview of Recent Trends.* https://medium.com/dair-ai/deep-learning-for-nlp-an-overview-of-recent-trends-d0d8f40a776d. Accessed: 2020/11/17.

[69] Danube Delta. *UNESCO World Heritage Centre.* http://whc.unesco.org/en/list/588. 2019.

[70] Yuri Demchenko, Paola Grosso, Cees De Laat, and Peter Membrey. "Addressing big data issues in scientific data infrastructure". In: *2013 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE. 2013, pp. 48–55.

[71] Janis L Dickinson, Jennifer Shirk, David Bonter, Rick Bonney, Rhiannon L Crain, Jason Martin, Tina Phillips, and Karen Purcell. "The current state of citizen science as a tool for ecological research and public engagement". In: *Frontiers in Ecology and the Environment* 10.6 (2012), pp. 291–297.

[72] Janis L. Dickinson, Benjamin Zuckerberg, and David N. Bonter. "Citizen Science as an Ecological Research Tool: Challenges and Benefits". In: *Annual Review of Ecology, Evolution, and Systematics* 41.1 (2010), pp. 149–172. DOI: 10.1146/annurev-ecolsys-102209-144636. URL: https://doi.org/10.1146/annurev-ecolsys-102209-144636.

[73] Jiri Dokulil and Jana Katreniaková. "Using Clusters in RDF Visualization". In: *Advances in Semantic Processing*. 2009.

[74] Hai Dong, Farookh Hussain, and Elizabeth Chang. "A context-aware semantic similarity model for ontology". In: *Concurrency and Computation: Practice and Experience* 23 (Apr. 2011), pp. 505–524. DOI: 10.1002/cpe.1652.

[75] Ralf Dörner, Stefan Göbel, Wolfgang Effelsberg, and Josef Wiemeyer. *Serious Games: Foundations, Concepts and Practice.* Jan. 2016. ISBN: 978-3-319-40611-4. DOI: 10.1007/978-3-319-40612-1.

[76] *DrugBank.* http://wifo5-03.informatik.uni-mannheim.de/drugbank/snorql/. 2019.

[77] Paul DuBois and Michael Foreword By-Widenius. *MySQL.* New riders publishing, 1999.

[78] Paul Ekman. "A methodological discussion of nonverbal behavior". In: *The Journal of psychology* 43.1 (1957), pp. 141–149.

[79] Paul Ekman and Dacher Keltner. "Universal facial expressions of emotion". In: *Segerstrale U, P. Molnar P, eds. Nonverbal communication: Where nature meets culture* (1997), pp. 27–46.

[80] Kareem El Gebaly and Jimmy Lin. "In-browser interactive SQL analytics with Afterburner". In: *ACM ICMD*. ACM. 2017.

[81] *El Viajero's tourism dataset*. `http://webenemasuno.linkeddata.es/sparql`. 2019.

[82] Jeffrey L Elman. "Finding structure in time". In: *Cognitive science* 14.2 (1990), pp. 179–211.

[83] *Emoji*. `https://github.com/carpedm20/emoji/`. Accessed: 2020/10/28.

[84] *Emoji Tracker*. `http://emojitracker.com/`. Accessed: 2020/11/17.

[85] Celia Evans, Eleanor Abrams, Robert Reitsma, Karin Roux, Laura Salmonsen, and Peter P Marra. "The Neighborhood Nestwatch Program: Participant outcomes of a citizen-science ecological research project". In: *Conservation Biology* 19.3 (2005), pp. 589–594.

[86] Celia Evans, Eleanor Abrams, Robert Reitsma, Karin Roux, Laura Salmonsen, and Peter P Marra. "The Neighborhood Nestwatch Program: Participant outcomes of a citizen-science ecological research project". In: *Conservation Biology* 19.3 (2005), pp. 589–594.

[87] *EventMedia*. `http://eventmedia.eurecom.fr/sparql`. 2019.

[88] Stephanie DH Evergreen. *Effective data visualization: The right chart for the right data*. Sage Publications, 2019.

[89] SeanM. Falconer, Chris Callendar, and Margaret-Anne Storey. "A Visualization Service for the Semantic Web". In: *Knowledge Engineering and Management by the Masses*. 2010.

[90] Wei Fan and Albert Bifet. "Mining big data: current status, and forecast to the future". In: *ACM sIGKDD Explorations Newsletter* 14.2 (2013), pp. 1–5.

[91] Atefeh Farzindar and Diana Inkpen. "Natural language processing for social media". In: *Synthesis Lectures on Human Language Technologies* 8.2 (2015), pp. 1–166.

[92] Ronen Feldman. "Techniques and applications for sentiment analysis". In: *Communications of the ACM* 56.4 (2013), pp. 82–89.

[93] Christiane Fellbaum. "WordNet". In: *The encyclopedia of applied linguistics* (2012).

[94] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. "bbuilding Watson: An Overview of the DeepQA Project". In: *AI Magazine* 31.3 (July 2010), pp. 59–79. DOI: `10.1609/aimag.v31i3.2303`. URL: `https://www.aaai.org/ojs/index.php/aimagazine/article/view/2303`.

[95] *Filter realtime Tweets*. `https://developer.twitter.com/en/docs/twitter-api/v1/tweets/filter-realtime/guides/basic-stream-parameters`. Accessed: 2020/11/17.

[96] Raphael A. Finkel and Jon Louis Bentley. "Quad trees a data structure for retrieval on composite keys". In: *Acta informatica* 4.1 (1974), pp. 1–9.

[97] Klaus Finkenzeller. *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication.* John wiley & sons, 2010.

[98] Edward W Forgy. "Cluster analysis of multivariate data: efficiency versus interpretability of classifications". In: *scentiometrics* 21 (1965), pp. 768–769.

[99] Lucy Fortson, Karen Masters, Robert Nichol, EM Edmondson, C Lintott, J Raddick, and J Wallin. "Galaxy zoo". In: *Advances in machine learning and data mining for astronomy* 2012 (2012), pp. 213–236.

[100] Santo Fortunato and Andrea Lancichinetti. "Community detection algorithms: a comparative analysis: invited presentation, extended abstract". In: *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools.* 2009, pp. 1–2.

[101] Henry Fuchs, Zvi M Kedem, and Bruce F Naylor. "On visible surface generation by a priori tree structures". In: *Proceedings of the 7th annual conference on Computer graphics and interactive techniques.* 1980, pp. 124–133.

[102] Mathias Fuchs, Sonia Fizek, Paolo Ruffino, and Niklas Schrape. *Rethinking gamification.* meson press, 2014.

[103] T. Fullerton. *Game Design Workshop: A Playcentric Approach to Creating Innovative Games.* Taylor & Francis, 2008. ISBN: 9780240809748. URL: `https://books.google.gr/books?id=0jIYWtqWxtAC`.

[104] Evgeniy Gabrilovich, Shaul Markovitch, et al. "Computing semantic relatedness using wikipedia-based explicit semantic analysis." In: *IJcAI.* Vol. 7. 2007, pp. 1606–1611.

[105] Amir Gandomi and Murtaza Haider. "bbeyond the hype: Big data concepts, methods, and analytics". In: *International journal of information management* 35.2 (2015), pp. 137–144.

[106] Mary M Gardiner, Leslie L Allee, Peter MJ Brown, John E Losey, Helen E Roy, and Rebecca Rice Smyth. "Lessons from lady beetles: accuracy of monitoring data from US and UK citizen-science programs". In: *Frontiers in Ecology and the Environment* 10.9 (2012), pp. 471–476.

[107] H. Geoghegan, A. Dyke, R. Pateman, S West, and G Everett. "Understanding Motivations for Citizen Science". In: *UK Environmental Observation Framework* (2016).

[108] GeoHash. `http://geohash.org`. 2020.

[109] *Geological Survey of Austria (GBA) - Thesaurus.* `https://resource.geolba.ac.at/PoolParty/sparql/lithology`. 2019.

[110] Geomesa. `https://www.geomesa.org/`. 2020.

[111] Paramita (Guha) Ghosh. *The Impact of Data Quality in the Machine Learning Era.* `https://www.dataversity.net/impact-data-quality-machine-learning-era/`. 2018.

[112] Michelle Girvan and Mark EJ Newman. "Community structure in social and biological networks". In: *Proceedings of the national academy of sciences* 99.12 (2002), pp. 7821–7826.

[113] *GraphViz*. `https://www.graphviz.org/`. 2018.

[114] Alvaro Graves. “Creation of visualizations based on linked data”. In: *International Conference on WIMS*. ACM. 2013.

[115] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. “Internet of Things (IoT): A vision, architectural elements, and future directions”. In: *Future generation computer systems* 29.7 (2013), pp. 1645–1660.

[116] P Gupta, P Doraiswamy, R Levy, O Pikelnaya, J Maibach, B Feenstra, Andrea Polidori, F Kiros, and KC Mills. “Impact of California Fires on Local and Regional Air Quality: The Role of a Low-Cost Sensor Network and Satellite Observations”. In: *GeoHealth* (2018).

[117] Antonin Guttman. “R-trees: A dynamic index structure for spatial searching”. In: *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*. 1984, pp. 47–57.

[118] H2database. `http://www.h2database.com/html/main.html`. 2020.

[119] H2gis. `http://www.h2gis.org/`. 2020.

[120] Florian Haag, Steffen Lohmann, and Thomas Ertl. “SparqlFilterFlow: SPARQL query composition for everyone”. In: *ESWC*. Springer. 2014.

[121] M Haklay. *Citizen Science and Policy: A European Perspective*. Washington, DC, USA: Woodrow Wilson International Center for Scholars, 2015.

[122] Guðmundur Jón Halldórsson. *Apache Accumulo for Developers*. Packt Publishing Ltd, 2013.

[123] Steve Harenberg, Gonzalo Bello, La Gjeltema, Stephen Ranshous, Jitendra Harlalka, Ramona Seay, Kanchana Padmanabhan, and Nagiza Samatova. “Community detection in large-scale networks: a survey and empirical evaluation”. In: Wiley Online Library, 2014.

[124] Jim Harris. *How data quality improves artificial intelligence*. `https://blogs.sas.com/content/datamanagement/2019/04/10/how-data-quality-improves-ai/`. 2019.

[125] Samer Hassan Hassan and Rada Mihalcea. “Semantic relatedness using salient semantic analysis”. In: *Twenty-Fifth AAAI Conference on Artificial Intelligence*. 2011.

[126] Tuukka Hastrup, Richard Cyganiak, and Uldis Bojars. “Browsing Linked Data with Fenfire”. In: *WWW*. 2008.

[127] Yaobin He, Haoyu Tan, Wuman Luo, Shengzhong Feng, and Jianping Fan. “MR-DBSCAN: a scalable MapReduce-based DBSCAN algorithm for heavily skewed data”. In: *Frontiers of Computer Science* 8.1 (2014), pp. 83–99.

[128] Philipp Heim, Thomas Ertl, and Jürgen Ziegler. “Facet graphs: Complex semantic querying made easy”. In: *ESWC*. Springer. 2010.

[129] Philipp Heim, Sebastian Hellmann, Jens Lehmann, Steffen Lohmann, and Timo Stegemann. “RelFinder: Revealing relationships in RDF knowledge bases”. In: *Semantic and Digital Media Technologies*. Springer. 2009.

[130] Philipp Heim, Steffen Lohmann, and Timo Stegemann. "Interactive Relationship Discovery via the Semantic Web". In: *ESWC*. 2010.

[131] Philipp Heim, Jürgen Ziegler, and Steffen Lohmann. "gFacet: A Browser for the Web of Data". In: *IMC-SSW'08*. Citeseer. 2008.

[132] Joseph M Hellerstein, Jeffrey F Naughton, and Avi Pfeffer. *Generalized search trees for database systems*. September, 1995.

[133] Jiří Helmich, Jakub Klímek, and Martin Nečaský. "Visualizing RDF data cubes using the linked data visualization model". In: *ESWC*. Springer. 2014.

[134] John R Herring. "Opengis implementation standard for geographic information-simple feature access-part 2: Sql option". In: *Open Geospatial Consortium Inc* (2010), p. 439.

[135] David Hilbert. "Über die stetige Abbildung einer Linie auf ein Flächenstück". In: *Dritter Band: Analysis· Grundlagen der Mathematik· Physik Verschiedenes*. Springer, 1935, pp. 1–2.

[136] Martin Hilbert. "bbig data for development: A review of promises and challenges". In: *Development Policy Review* 34.1 (2016), pp. 135–174.

[137] Martin Hilbert and Priscila López. "The World's Technological Capacity to Store, Communicate, and Compute Information". In: *Science (New York, N.Y.)* 332 (Feb. 2011), pp. 60–5. DOI: 10.1126/science.1200970.

[138] Sepp Hochreiter and Jurgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[139] Patrick Hoefler, Michael Granitzer, Eduardo E Veas, and Christin Seifert. "Linked Data Query Wizard: A Novel Interface for Accessing SPARQL Endpoints." In: *LDOW*. 2014.

[140] HRTA. *Hellenic Rescue Team of Attica*. http://www.eodathens.gr/. 2020.

[141] James N Hughes, Andrew Annex, Christopher N Eichelberger, Anthony Fox, Andrew Hulbert, and Michael Ronquest. "Geomesa: a distributed architecture for spatio-temporal fusion". In: *Geospatial Informatics, Fusion, and Motion Video Analytics V*. International Society for Optics and Photonics. 2015.

[142] James N Hughes, Andrew Annex, Christopher N Eichelberger, Anthony Fox, Andrew Hulbert, and Michael Ronquest. "Geomesa: a distributed architecture for spatio-temporal fusion". In: *Geospatial Informatics, Fusion, and Motion Video Analytics V*. Vol. 9473. International Society for Optics and Photonics. 2015, 94730F.

[143] *Human cap-dependent 48S pre-initiation complex*. http://www.rcsb.org/structure/6FEC. 2018.

[144] Sangyong Hwang, Keunjoo Kwon, Sang K. Cha, and Byung S. Lee. "Performance Evaluation of Main-Memory R-tree Variants". In: *Advances in Spatial and Temporal Databases*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 10–27. ISBN: 978-3-540-45072-6.

[145] Ioanna Iacovides, Charlene Jennett, Cassandra Cornish-Trestrail, and Anna L Cox. "Do games attract or sustain engagement in citizen science?: a study of volunteer motivations". In: *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. ACM. 2013, pp. 1101–1106.

[146] Spatial indices. http://www.h2gis.org/docs/dev/spatial-indices/. 2020.

[147] *Information about the Web-based Systems Group*. http://wifo5-03.informatik.uni-mannheim.de/dws-group/snorql/. 2019.

[148] *Interactive Visualization of Large Graphs*. Omitted for double-blind review. 2018.

[149] IoT. https://www.rfidjournal.com/that-internet-of-things-thing. 2020.

[150] *Isidore*. https://isidore.science/sparql. 2019.

[151] *Jamendo*. http://dbtune.org/jamendo/cliopatria/yasgui/index.html. 2019.

[152] Jay J Jiang and David W Conrath. "Semantic similarity based on corpus statistics and lexical taxonomy". In: *arXiv preprint cmp-lg/9709008* (1997).

[153] Jyun-Yu Jiang, Francine Chen, Yan-Ying Chen, and Wei Wang. "Learning to disentangle interleaved conversational threads with a siamese hierarchical network and similarity ranking". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2018, pp. 1812–1822.

[154] Tomasz Jóźwiak. "Tendencies in the numbers of beverage containers on the Polish coast in the decade from 1992 to 2001". In: *Marine Pollution Bulletin* 50.1 (2005), pp. 87–90. ISSN: 0025-326X. DOI: https://doi.org/10.1016/j.marpolbul.2004.11.011. URL: http://www.sciencedirect.com/science/article/pii/S0025326X04004333.

[155] José Fernando Rodrigues Jr., Hanghang Tong, Jia-Yu Pan, Agma J. M. Traina, Caetano Traina Jr., and Christos Faloutsos. "Large Graph Analysis in the GMine System". In: *TKDE* 25.1 (2013).

[156] José Fernando Rodrigues Jr., Hanghang Tong, Agma J. M. Traina, Christos Faloutsos, and Jure Leskovec. "GMine: A System for Scalable, Interactive Graph Visualization and Mining". In: *VLDB*. 2006.

[157] Dan Jurafsky. *Speech & language processing*. Pearson Education India, 2000.

[158] Tomihisa Kamada and Satoru Kawai. "An algorithm for drawing general undirected graphs". In: *Information Processing Letters* 31.1 (1989), pp. 7–15. ISSN: 0020-0190. DOI: https://doi.org/10.1016/0020-0190(89)90102-6. URL: http://www.sciencedirect.com/science/article/pii/0020019089901026.

[159] Ibrahim Kamel and Christos Faloutsos. *Hilbert R-tree: An Improved R-tree using Fractals*. June 2018. DOI: 10.1184/R1/6606125.v1. URL: https://kilthub.cmu.edu/articles/journal%5C_contribution/Hilbert%5C_R-tree%5C_An%5C_Improved%5C_R-tree%5C_using%5C_Fractals/6606125/1.

[160] Zahra Karevan and Johan AK Suykens. "Transductive LSTM for time-series prediction: An application to weather forecasting". In: *Neural Networks* 125 (2020), pp. 1–9.

[161] George Karypis and Vipin Kumar. "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs". In: *SIAM J. Sci. Comput.* 20.1 (Dec. 1998), pp. 359–392. ISSN: 1064-8275. DOI: 10.1137/S1064827595287997. URL: http://dx.doi.org/10.1137/S1064827595287997.

[162] Pond Kathy and Rees Gareth. "Coastwatch UK—a public participation survey". In: *Journal of Coastal Conservation* 6.1 (Dec. 2000), pp. 61–66. ISSN: 1874-7841. DOI: 10.1007/BF02730469. URL: https://doi.org/10.1007/BF02730469.

[163] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis.* Vol. 344. John Wiley & Sons, 2009.

[164] Edric Keighan, Panagiotis A Vretanos, Michael Galluchon, and Herman P Varma. *Method and apparatus for multidimensional database using binary hyperspatial code.* US Patent 6,161,105. Dec. 2000.

[165] Christa Kelleher and Thorsten Wagener. "Ten guidelines for effective data visualization in scientific publications". In: *Environmental Modelling & Software* 26 (2011).

[166] Dr Andreas Oskar Kempf. *STW-INFO.* http://www.zbw.eu/en/stw-info/. 2019.

[167] R Kerson. *Lab for the Environment.* Vol. 92. MIT Technology Review, 1989, pp. 11–12.

[168] Ashraf M Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. "Multinomial naive bayes for text categorization revisited". In: *Australasian Joint Conference on Artificial Intelligence.* Springer. 2004, pp. 488–499.

[169] Younghoon Kim, Kyuseok Shim, Min-Soeng Kim, and June Sup Lee. "DBCURE-MR: An efficient density-based clustering algorithm for large data using MapReduce". In: *Information Systems* 42 (2014), pp. 15–35.

[170] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[171] Jakub Klímek, Jirí Helmich, and Martin Necaský. "Payola: Collaborative Linked Data Analysis and Visualization Framework". In: *ESWC.* 2013.

[172] Margaret Kosmala, Andrea Wiggins, Alexandra Swanson, and Brooke Simmons. "Assessing data quality in citizen science". In: *Frontiers in Ecology and the Environment* 14 (Dec. 2016), pp. 551–560. DOI: 10.1002/fee.1436.

[173] Alexander Kotsev, Sven Schade, Massimo Craglia, Michel Gerboles, Laurent Spinelle, and Marco Signorini. "Next Generation Air Quality Platform: Openness and Interoperability for the Internet of Things". In: *Sensors* 16.3 (2016). DOI: 10.3390/s16030403.

[174] George Krasadakis. *Data quality in the era of Artificial Intelligence.* https://medium.com/ideachain/data-quality-in-the-era-of-a-i-d8e398a91bef. 2017.

[175] M. Krommyda and V. Kantere. "Improving the Quality of the Conversational Datasets through Extensive Semantic Analysis". In: *2019 IEEE International Conference on Conversational Data Knowledge Engineering (CDKE).* 2019, pp. 1–8.

[176] M. Krommyda and V. Kantere. “Semantic analysis for conversational datasets: improving their quality using semantic relationships”. In: *International Journal of Semantic Computing* 14.3 (2020).

[177] Maria Krommyda, Anastatios Rigos, Kostas Bouklas, and Angelos Amditis. “Emotion detection in Twitter posts: a rule-based algorithm for annotated data acquisition”. In: *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE. 2020.

[178] Randy Krum. *Cool infographics: Effective communication with data visualization and design*. John Wiley & Sons, 2013.

[179] Auto-id labs. `https://www.autoidlabs.org/`. 2020.

[180] Anne M. Land-Zandstra, Jeroen L. A. Devilee, Frans Snik, Franka Buurmeijer, and Jos M. van den Broek. “Citizen science on a smartphone: Participants’ motivations and learning”. In: *Public Understanding of Science* 25.1 (2016), pp. 45–60. DOI: `10.1177/0963662515602406`. URL: `https://doi.org/10.1177/0963662515602406`.

[181] Mark E Larsen, Tjeerd W Boonstra, Philip J Batterham, Bridianne O’Dea, Cecile Paris, and Helen Christensen. “We feel: mapping emotion on Twitter”. In: *IEEE journal of biomedical and health informatics* 19.4 (2015), pp. 1246–1252.

[182] Alexander de Leon, Filip Wisniewki, Boris Villazón-Terrazas, and Oscar Corcho. “Map4rdf- Faceted Browser for Geospatial Datasets”. In: *Using Open Data: policy modeling, citizen empowerment, data journalism*. 2012.

[183] Steve Liang, Chih-Yuan Huang, and Tania Khalafbeigi. “OGC SensorThings API Part 1: Sensing, Version 1.0.” In: (2016).

[184] DR Liberman. “Codd’s 12 rules: A method for DBMS evaluation.” In: *DATABASE PROGRAM. DES.* 1.12 (1988), pp. 30–38.

[185] *Linked Open Data*. `https://datahub.io/collections/linked-open-data`. 2018.

[186] *Linked Open Data Camera dei deputati*. `http://dati.camera.it/sparql`. 2019.

[187] *Linked Open Vocabularies (LOV)*. `https://lov.linkeddata.es/dataset/lov/sparql`. 2019.

[188] *Lista de Encabezamientos de Materia as Linked Open*. `http://id.sgcb.mcu.es/sparql`. 2019.

[189] Bing Liu. “Sentiment analysis and opinion mining”. In: *Synthesis lectures on human language technologies* 5.1 (2012), pp. 1–167.

[190] Stuart Lloyd. “Least squares quantization in PCM”. In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137.

[191] Steffen Lohmann, Stefan Negru, Florian Haag, and Thomas Ertl. “Visualizing Ontologies with VOWL”. In: *Semantic Web Journal* (2015).

[192] Lu Tan and Neng Wang. “Future internet: The Internet of Things”. In: *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*. Vol. 5. 2010, pp. V5–376-V5–380.

[193] Kurt Luther, Scott Counts, Kristin B Stecher, Aaron Hoff, and Paul Johns. “Pathfinder: an misc collaboration environment for citizen scientists”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2009, pp. 239–248.

[194] Gillian C Lye, Juliet L Osborne, Kirsty J Park, and Dave Goulson. “Using citizen science to monitor Bombus populations in the UK: nesting ecology and relative abundance in the urban environment”. In: *Journal of Insect Conservation* 16.5 (2012), pp. 697–707.

[195] *Magnatune from DBTune*. `http://dbtune.org/magnatune/cliopatria/yasgui/index.html`. 2019.

[196] Ana G Maguitman, Filippo Menczer, Heather Roinestad, and Alessandro Vespignani. “Algorithmic detection of semantic similarity”. In: *Proceedings of the 14th international conference on World Wide Web*. ACM. 2005, pp. 107–116.

[197] Stanislav Malyshev, Markus Krötzsch, Larry González, Julius Gonsior, and Adrian Bielefeldt. “Getting the Most out of Wikidata: Semantic Technology Usage in Wikipedia’s Knowledge Graph”. In: *ISWC 18*. Springer, 2018.

[198] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[199] James Manyika. “Big data: The next frontier for innovation, competition, and productivity”. In: (2011). URL: `http://www.%20mckinsey.%20com/Insights/MGI/Research/Technology%5C_and%5C_Innovation/%20Big%5C_data%5C_The%5C_next%5C_frontier%5C_for%5C_innovation`.

[200] N Justin Marshall, Diana A Kleine, and Angela J Dean. “CoralWatch: education, monitoring, and sustainability through citizen science”. In: *Frontiers in Ecology and the Environment* 10.6 (2012), pp. 332–334.

[201] Martin Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: `http://tensorflow.org/`.

[202] Suvodeep Mazumdar, Daniela Petrelli, Khadija Elbedweihy, Vitaveska Lanfranchi, and Fabio Ciravegna. “Affective graphs: The visual appeal of Linked Data”. In: *Semantic Web* 6.3 (2015).

[203] Donald JR Meagher. *Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer*. Electrical and Systems Engineering Department Rensseiaer Polytechnic ..., 1980.

[204] David Mercer. “Global Connected and IoT Device Forecast Update”. In: *Tech. rep. Strategy Analytics* (2019).

[205] *METIS*. `http://glaros.dtc.umn.edu/gkhome/metis/metis/overview`. 2018.

[206] Kai-chao Miao, Ting-ting Han, Ye-qing Yao, Hui Lu, Peng Chen, Bing Wang, and Jun Zhang. “Application of LSTM for Short Term Fog Forecasting based on Meteorological Elements”. In: *Neurocomputing* (2020).

[207] Trevor van Mierlo. “The 1% Rule in Four Digital Health Social Networks: An Observational Study”. In: *J Med Internet Res* 16.2 (Feb. 2014), e33. ISSN: 14388871. DOI: 10.2196/jmir.2966.

[208] Rada Mihalcea, Courtney Corley, Carlo Strapparava, et al. “Corpus-based and knowledge-based measures of text semantic similarity”. In: *Aaai*. Vol. 6. 2006. 2006, pp. 775–780.

[209] Justin J Miller. “Graph database applications and concepts with Neo4j”. In: *Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA*. Vol. 2324. 36. 2013.

[210] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. “Internet of things: Vision, applications and research challenges”. In: *Ad Hoc Networks* 10.7 (2012), pp. 1497–1516. ISSN: 1570-8705. DOI: https://doi.org/10.1016/j.adhoc.2012.02.016. URL: http://www.sciencedirect.com/science/article/pii/S1570870512000674.

[211] Volker Mische. “GeoCouch: A spatial index for CouchDB”. In: *Presentation at FOSS5G* (2010).

[212] Saif M Mohammad and Peter D Turney. “Nrc emotion lexicon”. In: *National Research Council, Canada* 2 (2013).

[213] Andrew W Moore. “An intoductory tutorial on kd-trees”. In: (1991).

[214] Guy M Morton. “A computer oriented geodetic data base and a new technique in file sequencing”. In: (1966).

[215] Enrico Motta, Paul Mulholland, Silvio Peroni, Mathieu d’Aquin, José Manuél Gómez-Pérez, Victor Mendez, and Fouad Zablith. “A Novel Approach to Visualizing and Navigating Ontologies”. In: *ISWC*. 2011.

[216] Conor Murphy, Robert L. Wilby, K.R. Matthews Tom, Peter Thorne, Ciaran Broderick, Rowan Fealy, Julia Hall, Shaun Harrigan, Phil Jones, Gerard McCarthy, Neil Macdonald, Simon Noone, and Ciara Ryan. “Multi-century trends to wetter winters and drier summers in the England and Wales precipitation series explained by observational and sampling bias in early records”. In: *International Journal of Climatology* (2019). DOI: 10.1002/joc.6208.

[217] J Mustafa and Marios D Dikaiakos. “MashQL: A Query-by-Diagram Topping SPARQL Towards Semantic Data Mashups”. In: *University of Cyprus mjarrar* ().

[218] MySQL. https://dev.mysql.com/doc/refman/8.0/en/creating-spatial-indexes.html. 2020.

[219] Ravi Narasimhan and T Bhuvaneshwari. “bbig data–a brief study”. In: *Int. J. Sci. Eng. Res* 5.9 (2014), pp. 350–353.

[220] Neo4j. https://neo4j.com/. 2020.

[221] Neo4j. https://neo4j.com/docs/cypher-manual/current/syntax/spatial/\#cypher-spatial-index. 2020.

[222] Martin E Newell. “A new approach to the shaded picture problem”. In: *Proc. ACM National Conf.* 1972.

[223] Greg Newman, Alycia Crall, Melinda Laituri, Jim Graham, Tom Stohlgren, John C Moore, Kris Kodrich, and Kirstin A Holfelder. “Teaching citizen science skills misc: Implications for invasive species training programs”. In: *Applied Environmental Education and Communication* 9.4 (2010), pp. 276–286.

[224] Greg Newman, Andrea Wiggins, Alycia Crall, Eric Graham, Sarah Newman, and Kevin Crowston. “The future of citizen science: emerging technologies and shifting paradigms”. In: *Frontiers in Ecology and the Environment* 10.6 (2012), pp. 298–304.

[225] Mark EJ Newman. “Modularity and community structure in networks”. In: *Proceedings of the national academy of sciences* 103.23 (2006), pp. 8577–8582.

[226] ALEX NGUYEN. *15 Best Chatbot Datasets for Machine Learning*. https://lionbridge.ai/datasets/15-best-chatbot-datasets-for-machine-learning/. 2019.

[227] Jakob Nielsen. *The 90-9-1 Rule for Participation Inequality in Social Media and misc Communities*. https://www.nngroup.com/articles/participation-inequality/. 2016.

[228] *NLTP Corpus*. http://www.nltk.org/howto/corpus.html. Accessed: 2020/11/17.

[229] Jakub Nowak, Ahmet Taspinar, and Rafał Scherer. “LSTM recurrent neural networks for short text and sentiment classification”. In: *International Conference on Artificial Intelligence and Soft Computing*. Springer. 2017, pp. 553–562.

[230] Regina Obe and Leo Hsu. “PostGIS in action”. In: *GEOInformatics* 14.8 (2011), p. 30.

[231] OGC. https://www.ogc.org/. 2020.

[232] OGC. *OGC Implementing/Compliant Product Details*. https://www.opengeospatial.org/resource/products/details/?pid=1554. 2019.

[233] OGC. *Welcome to The Open Geospatial Consortium*. https://www.opengeospatial.org. 2019.

[234] Frank J Ohlhorst. *bbig data analytics: turning big data into big money*. Vol. 65. John Wiley & Sons, 2012.

[235] Christopher Olah. “Understanding lstm networks”. In: *http://colah.github.io/posts/2015-08-Understanding-LSTMs* (2015).

[236] *Open Data Thesaurus*. http://vocabulary.semantic-web.at/PoolParty/sparql/OpenData. 2019.

[237] *Open Mobile Network*. http://www.openmobilenetwork.org:8890/sparql. 2019.

[238] *OpenLink Virtuoso*. http://demo.openlinksw.com/sparql/. 2019.

[239] Marcus Oswald, Gerhard Reinelt, and Stefan Wiesberg. “Exact solution of the 2-dimensional grid arrangement problem”. In: *Discrete Optimization* (2012).

[240] Gwen Ottinger. “scentuckets of resistance: Standards and the effectiveness of citizen science”. In: *Science, Technology, & Human Values* 35.2 (2010), pp. 244–270.

[241] *OxPoints (University of Oxford).* `https://data.ox.ac.uk/sparql/`. 2019.

[242] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[243] Viktor Pekar and Steffen Staab. “Taxonomy Learning: Factoring the Structure of a Taxonomy into a Semantic Classification Decision”. In: *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*. COLING ’02. Taipei, Taiwan: Association for Computational Linguistics, 2002, pp. 1–7. DOI: `10.3115/1072228.1072318`. URL: `https://doi.org/10.3115/1072228.1072318`.

[244] Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. “Align, disambiguate and walk: A unified approach for measuring semantic similarity”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1. 2013, pp. 1341–1351.

[245] Robert Plutchik. “A general psychoevolutionary theory of emotion”. In: *Theories of emotion*. Elsevier, 1980, pp. 3–33.

[246] Robert Plutchik. *The emotions*. University Press of America, 1991.

[247] Postgis. `https://postgis.net/stuff/postgis-3.0.pdf`. 2020.

[248] Jennifer Preece. “Citizen Science: New Research Challenges for Human–Computer Interaction”. In: *International Journal of Human–Computer Interaction* 32.8 (2016), pp. 585–612. DOI: `10.1080/10447318.2016.1194153`.

[249] Paul Ramsey and Victoria–British Columbia. “Introduction to postgis”. In: (2005).

[250] Adithya Rao and Nemanja Spasojevic. “Actionable and political text classification using word embeddings and lstm”. In: *arXiv preprint arXiv:1607.02501* (2016).

[251] Guozheng Rao, Weihang Huang, Zhiyong Feng, and Qiong Cong. “LSTM with sentence representations for document-level sentiment classification”. In: *Neurocomputing* 308 (2018), pp. 49–57.

[252] Responsible Conduct of Research RCR. *Data collection*. `https://ori.hhs.gov/education/products/n\_illinois\_u/datamanagement/dctopic.html`. 2019.

[253] Hope Reese. *Why Microsoft’s ’Tay’ AI bot went wrong*. `https://www.techrepublic.com/article/why-microsofts-tay-ai-bot-went-wrong/`. 2016.

[254] Philip Resnik. *Using Information Content to Evaluate Semantic Similarity in a Taxonomy*. 1995. arXiv: `cmp-lg/9511007 [cmp-lg]`.

[255] RethinkDB. `https://rethinkdb.com/docs/geo-support/python/`. 2020.

[256] *Revyu*. http://revyu.com/sparql/queryform. 2019.

[257] Stephen V. Rice, Frank Robert Jenkins, and Thomas A. Nartker. "The Fourth Annual Test of OCR Accuracy". In: 1995.

[258] Laurens Rietveld, Rinke Hoekstra, et al. "Man vs. machine: Differences in SPARQL queries". In: *ESWC*. 2014.

[259] Kifisos River. *Scent Project EU*. https://scent-project.eu/kifisos-river-basin-attica-greece. 2019.

[260] Joshua Roesslein. "Tweepy: Twitter for Python!" In: *URL: https://github.com/tweepy/tweepy* (). Accessed: 2020/11/17.

[261] Ira J Roseman. "Cognitive determinants of emotion: A structural theory." In: *Review of personality & social psychology* (1984).

[262] CASEY ROSS. *IBM's Watson supercomputer recommended 'unsafe and incorrect' cancer treatments, internal documents show*. https://www.statnews.com/2018/07/25/ibm-watson-recommended-unsafe-incorrect-treatments/. 2018.

[263] Dana Rotman, Jenny Preece, Jen Hammock, Kezee Procita, Derek Hansen, Cynthia Parr, Darcy Lewis, and David Jacobs. "Dynamic Changes in Motivation in Collaborative Citizen-science Projects". In: *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*. CSCW '12. Seattle, Washington, USA: ACM, 2012, pp. 217–226. ISBN: 978-1-4503-1086-4. DOI: 10.1145/2145204.2145238. URL: http://doi.acm.org/10.1145/2145204.2145238.

[264] Alaa Sagheer and Mostafa Kotb. "Time series forecasting of petroleum production using deep LSTM recurrent networks". In: *Neurocomputing* 323 (2019), pp. 203–213.

[265] Salvatore Sanfilippo and Pieter Noordhuis. *Redis*. 2009.

[266] Anne scentowser, Derek Hansen, Yurong He, Carol Boston, Matthew Reid, Logan Gunnell, and Jennifer Preece. "Using gamification to inspire new citizen science volunteers". In: *Proceedings of the first international conference on gameful design, research, and applications*. ACM. 2013, pp. 18–25.

[267] Roberto scentugiolacchi, Steven Bamford, Paul Tar, Neil Thacker, Ian A Crawford, Katherine H Joy, Peter M Grindrod, and Chris Lintott. "The Moon Zoo citizen science project: Preliminary results for the Apollo 17 landing site". In: *Icarus* 271 (2016), pp. 30–48.

[268] Jesse Schell. *The Art of Game Design: A Book of Lenses*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2008. ISBN: 0-12-369496-5.

[269] Kai Schlegel, Thomas Weißgerber, Florian Stegmaier, Christin Seifert, Michael Granitzer, and Harald Kosch. "Balloon Synopsis: A Modern Node-Centric RDF Viewer and Browser for the Web". In: *ESWC*. 2014.

[270] R Schumacher. *Study for applying computer-generated images to visual simulation*. Vol. 69. 14. Air Force Human Resources Laboratory, Air Force Systems Command, 1969.

[271] Linda See, Alexis Comber, Carl Salk, Steffen Fritz, Marijn van der Velde, Christoph Perger, Christian Schill, Ian McCallum, Florian Kraxner, and Michael Obersteiner. "Comparing the Quality of Crowdsourced Data Contributed by Expert and Non-Experts". In: *PLOS ONE* 8.7 (July 2013), pp. 1–11. DOI: 10.1371/journal.pone.0069958. URL: https://doi.org/10.1371/journal.pone.0069958.

[272] Timos Sellis, Nick Roussopoulos, and Christos Faloutsos. *The R+-Tree: A Dynamic Index for Multi-Dimensional Objects.* Tech. rep. 1987.

[273] *semanticSBML 2.0.* http://semanticsbml.org/semanticSBML/simple/index. 2018.

[274] SQL Server. https://www.microsoft.com/en-us/sql-server. 2020.

[275] SQL Server. https://docs.microsoft.com/en-us/sql/relational-databases/spatial/spatial-data-types-overview?view=sql-server-ver15. 2020.

[276] SQL Server. https://docs.microsoft.com/en-us/sql/relational-databases/spatial/spatial-indexes-overview?view=sql-server-ver15. 2020.

[277] SFS. http://www.opengeospatial.org/standards/sfs. 2020.

[278] Jonathan Silvertown. "A new dawn for citizen science". In: *Trends in ecology & evolution* 24.9 (2009), pp. 467–471.

[279] Ingo Simonis. "Standardized Information Models to Optimize Exchange, Reusability and Comparability of Citizen Science Data. A Specialization Approach". In: *International Journal of Spatial Data Infrastructures Research* 13 (2018), pp. 38–47.

[280] Robert Simpson, Kevin R Page, and David De Roure. "Zooniverse: observing the world's largest citizen science platform". In: *Proceedings of the 23rd international conference on world wide web*. ACM. 2014, pp. 1049–1054.

[281] R. Smith. "An Overview of the Tesseract OCR Engine". In: *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007).* Vol. 2. Sept. 2007, pp. 629–633.

[282] *Social Semantic Web Thesaurus.* http://vocabulary.semantic-web.at/PoolParty/sparql/semweb. 2019.

[283] Akila Somasundaram and U. Srinivasulu Reddy. "Data Imbalance: Effects and Solutions for Classification of Large and Highly Imbalanced Data". In: Jan. 2016.

[284] SOR. *Societatea Ornitologica Romana.* https://www.sor.ro/. 2020.

[285] IBM Spatial. https://www.ibm.com/support/knowledgecenter/SSGU8G11.50.0/com.ibm.gsg.doc/idsgsg260.htm. 2020.

[286] IBM Spatial. https://www.ibm.com/support/knowledgecenter/SSGU8G11.50.0/com.ibm.spatial.doc/sii-overview-10593.htm. 2020.

[287] Oracle Spatial and Graph. https://www.oracle.com/technetwork/database-options/spatialandgraph/overview/spatialandgraph-1707409.html. 2020.

[288] Claus Stadler, Michael Martin, and Sören Auer. "Exploring the web of spatial data with facete". In: *WWW*. 2014.

[289] Mark C. Stone and Rollin H. Hotchkiss. "Evaluating velocity measurement techniques in shallow streams". In: *Journal of Hydraulic Research* 45.6 (2007), pp. 752–762. DOI: `10.1080/00221686.2007.9521813`.

[290] Michael Stonebraker and Lawrence A Rowe. "The design of POSTGRES". In: *ACM Sigmod Record* 15.2 (1986), pp. 340–355.

[291] Magnus Stuhr, Dumitru Roman, and David Norheim. "LODWheel - JavaScript-based Visualization of RDF Data". In: *COLD*. 2011.

[292] *STW Thesaurus for Economics*. `http://zbw.eu/beta/sparql-lab/`. 2019.

[293] Brian L Sullivan, Christopher L Wood, Marshall J Iliff, Rick E Bonney, Daniel Fink, and Steve Kelling. "eBird: A citizen-based bird observation network in the biological sciences". In: *scentiological Conservation* 142.10 (2009), pp. 2282–2292.

[294] Seema Sundara, Medha Atre, Vladimir Kolovski, Souripriya Das, Zhe Wu, Eugene Inseok Chong, and Jagannathan Srinivasan. "Visualizing large-scale RDF data using Subsets, Summaries, and Sampling in Oracle". In: *ICDE*. 2010.

[295] Harald Sundmaeker, Patrick Guillemin, Peter Friess, and Sylvie Woelfflé. "Vision and challenges for realising the Internet of Things". In: *Cluster of European Research Projects on the Internet of Things, European Commision* 3.3 (2010), pp. 34–36.

[296] Nathan Sykes. *What To Know About The Impact of Data Quality and Quantity In AI*. `https://www.smartdatacollective.com/what-to-know-about-impact-data-quality-quantity-in-ai/`. 2018.

[297] *TaskForces/CommunityProjects/LinkingOpenData/DataSets/Statistics*. `https://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/DataSets/Statistics`. 2018.

[298] Jeff R Taylor and Henry L Loescher. "Automated quality control methods for sensor data: a novel observatory approach". In: *scentiogeosciences* 10 (2013).

[299] Technative. *Data Quality vs Data Quantity: What's More Important for AI?* `https://www.technative.io/data-quality-vs-data-quantity-whats-more-important-for-ai/`. 2018.

[300] *The 500 Most Frequently Used Words on Twitter*. `https://techland.time.com/2009/06/08/the-500-most-frequently-used-words-on-twitter/`. Accessed: 2020/11/17.

[301] Klaudia Thellmann, Mikhail Galkin, Fabrizio Orlandi, and Sören Auer. "LinkDaViz - Automatic Binding of Linked Data to Visualizations". In: *ISWC*. 2015.

[302] Yuanyuan Tian, Richard A Hankins, and Jignesh M Patel. "Efficient aggregation for graph summarization". In: *ACM SIGMOD*. 2008.

[303] Gianluca Tiepolo. *Getting started with rethinkdb*. Packt Publishing Ltd, 2016.

[304] *Time Ontology in OWL*. `https://www.w3.org/TR/owl-time/`. 2019.

[305] Silvan S Tomkins. *Affect imagery consciousness: Volume I: The positive affects*. Vol. 1. Springer publishing company, 1962.

[306] Brian Trench. "Science communication and citizen science: how dead is the deficit model". In: *IX International Conference on Public Comunication of Science and Technology (PCST), Seoul, Korea*. 2006.

[307] A. Tserstou, A. Jonoski, I. Popescu, T. H. Asumpcao, G. Athanasiou, A. Kallioras, and I. Nichersu. "SCENT: Citizen Sourced Data in Support of Environmental Monitoring". In: *2017 21st International Conference on Control Systems and Computer Science (CSCS)*. 2017, pp. 612–616.

[308] Edward R Tufte. *The visual display of quantitative information*. Vol. 2. Graphics press Cheshire, CT, 2001.

[309] *Tweet Preprocessor*. `https://pypi.org/project/tweet-preprocessor/`. Accessed: 2020/11/17.

[310] *Twitter*. `https://twitter.com/home?lang=en`. Accessed: 2020/11/17.

[311] *Twitter Developer Docs*. `https://developer.twitter.com/en/docs`. Accessed: 2020/11/17.

[312] Muhammad Fahim Uddin, Navarun Gupta, et al. "Seven V's of Big Data understanding Big Data to extract value". In: *Proceedings of the 2014 zone 1 conference of the American Society for Engineering Education*. IEEE. 2014, pp. 1–5.

[313] *UN Comtrade Database*. `https://comtrade.un.org/`. 2020.

[314] *UniProt*. `https://sparql.uniprot.org/sparql`. 2019.

[315] Princeton University. *WordNet: A Lexical Database for English*. `https://wordnet.princeton.edu/`. 2019.

[316] *URIBurner.com*. `http://uriburner.com/sparql/`. 2019.

[317] *Vacancies (University of Oxford)*. `https://data.ox.ac.uk/sparql/`. 2019.

[318] Herman Varma, H Boudreau, and W Prime. "A data structure for spatio-temporal databases". In: *International Hydrographic Review* 67.1 (1990), pp. 71–92.

[319] Ronny Verhoeven, Robert Banasiak, Tomasz Okruszko, D Światek, Jaroslaw Chormanski, P Nowakowski, Ignacy Kardel, and M Stelmaszczyk. "Hydraulic Modelling of River Flow - data collection and problem solving". In: (Jan. 2003).

[320] Luc Vincent. *Announcing Tesseract OCR*. `http://googlecode.blogspot.com/2006/08/announcing-tesseract-ocr.html`. 2006.

[321] Martin Voigt, Stefan Pietschmann, Lars Grammel, and Klaus Meißner. "Context-aware Recommendation of Visualization Components". In: *eKNOW*. 2012.

[322] Jin Wang, Bo Peng, and Xuejie Zhang. "Using a stacked residual LSTM model for sentiment intensity prediction". In: *Neurocomputing* 322 (2018), pp. 93–101.

[323] Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P Sheth. “Harnessing twitter" big data" for automatic emotion identification”. In: *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*. IEEE. 2012, pp. 587–592.

[324] Yafang Wang, Mingjie Zhu, Lizhen Qu, Marc Spaniol, and Gerhard Weikum. “Timely yago: harvesting, querying, and visualizing temporal knowledge from wikipedia”. In: *International Conference on Extending Database Technology*. ACM. 2010.

[325] Jonathan Stuart Ward and Adam Barker. “Undefined by data: a survey of big data definitions”. In: *arXiv preprint arXiv:1309.5821* (2013).

[326] Jim Webber. “A programmatic introduction to neo4j”. In: *Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity*. 2012, pp. 217–218.

[327] Marc Weise, Steffen Lohmann, and Florian Haag. “Extraction and visualization of tbox information from sparql endpoints”. In: *EKAW*. Springer. 2016.

[328] Bernard Lewis Welch. “On the comparison of several mean values: an alternative approach”. In: *scentiometrika* 38.3-4 (1951), pp. 330–336.

[329] Etienne Wenger. *Communities of practice: Learning, meaning, and identity*. Cambridge university press, 1999.

[330] Michael Widenius, David Axmark, and Kaj Arno. *MySQL reference manual: documentation from the source*. " O'Reilly Media, Inc.", 2002.

[331] *Wikipedia:Statistics*. `https://en.wikipedia.org/wiki/Wikipedia:Statistics`. 2020.

[332] Nathan Willis. *Google's Tesseract OCR engine is a quantum leap forward*. `https://www.linux.com/news/googles-tesseract-ocr-engine-quantum-leap-forward`. 2006.

[333] Leibniz-Informationszentrum Wirtschaft. *ZBW*. `http://www.zbw.eu/de/`. 2019.

[334] Ian H Witten and David N Milne. “An effective, low-cost measure of semantic relatedness obtained from Wikipedia links”. In: (2008).

[335] Xiaomi. *Huahuacaocao Flower Care Smart Monitor*. `https://xiaomi-mi.com/sockets-and-sensors/xiaomi-huahuacaocao-flower-care-smart-monitor/`. 2020.

[336] Atsuko Yamaguchi, Kouji Kozaki, Kai Lenz, Hongyan Wu, and Norio Kobayashi. “An Intelligent SPARQL Query Builder for Exploration of Various Life-science Databases.” In: *IESD ISWC*. 2014.

[337] Thomas H Yorke and Kevin A Oberg. “Measuring river velocity and discharge with acoustic Doppler profilers”. In: *Flow Measurement and Instrumentation* 13.5 (2002), pp. 191–195. ISSN: 0955-5986. DOI: `https://doi.org/10.1016/S0955-5986(02)00051-1`.

[338] JH Zar. *scentiostatistical Analysis 5th ed New York*. 1998.

[339] Kang Zhang, Haofen Wang, Duc Thanh Tran, and Yong Yu. “ZoomRDF: semantic fisheye zooming on RDF data”. In: *WWW*. 2010.

[340] Ying Zhu. “Measuring effective data visualization”. In: *International Symposium on Visual Computing*. Springer. 2007.

[341] Michael Zinsmaier, Ulrik Brandes, Oliver Deussen, and Hendrik Strobelt. “Interactive Level-of-Detail Rendering of Large Graphs”. In: *TVCG* 18.12 (2012).

[342] Zhen Zuo. “Sentiment analysis of steam review datasets using naive bayes and decision tree classifier”. In: (2018).

[343] Martins Zviedris and Guntis Barzdins. “ViziQuer: a tool to explore and query SPARQL endpoints”. In: *ESWC*. Springer. 2011.

Author Publications

[1] Anna Antonakopoulou, Evangelia Portouli, Nikolaos Tousert, Maria Krommyda, Angelos Amditis, Maria Pia Fanti, Alessandro Rinaldi, and Bartolomeo Silvestri. “Accelerating the Deployment of Electric Light Vehicles for Sustainable Urban Mobility: A Harmonized Pilot Demonstration Methodology”. In: *Advances in Mobility-as-a-Service Systems*. Ed. by Eftihia G. Nathanail, Giannis Adamos, and Ioannis Karakikes. Cham: Springer International Publishing, 2021, pp. 181–191.

[2] Maria K Krommyda and Verena Kantere. “The Big Data Era: Data Management Novelties for Visualizing, Exploring, and Processing Big Data”. In: *Analyzing Future Applications of AI, Sensors, and Robotics in Society*. IGI Global, 2021, pp. 87–103.

[3] Rigos Anastasios, Krommyda Maria, Theodoropoulos Theodoros, Tsiakos Valantis, Tsertou Athanasia, and Amditis Angelos. “Collecting and Processing Crowdsourced River Measurements Using Innovative Sensors”. In: *Science Signpost Publishing Inc* (2020).

[4] M. Krommyda and V. Kantere. “A Framework for Exploration and Visualization of SPARQL Endpoint Information”. In: *International Journal of Graph Computing* 1.1 (2020), pp. 39–69. DOI: 10.35708/GC1868-126723.

[5] M. Krommyda and V. Kantere. “Semantic analysis for conversational datasets: improving their quality using semantic relationships”. In: *International Journal of Semantic Computing* 14.3 (2020).

[6] M. Krommyda and V. Kantere. “Spatial Data Management in IoT systems: A study of available storage and indexing solutions”. In: *2020 Second International Conference on Transdisciplinary AI (TransAI)*. 2020, pp. 146–153. DOI: 10.1109/TransAI49837.2020.00033.

[7] M. Krommyda and V. Kantere. “Spatial Data Management in IoT systems: Solutions & Evaluation.” In: *International Journal of Semantic Computing* (2020).

[8] M. Krommyda and V. Kantere. “Visualization Systems for Linked Datasets”. In: *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. 2020, pp. 1790–1793.

[9] Maria Krommyda, Richardos Drakoulis, Fay Misichroni, Nikolaos Tousert, Anna Antonakopoulou, Evangelia Portouli, Mandimby N. Ranaivo Rakotondravelona, Marwane El-Bekri, Djibrilla Amadou Kountché, and Angelos Amditis. “An ICT Platform for the Understanding of the User Behaviours towards EL-Vs”. In: *Proceedings of the 22nd International Conference on Enterprise Information Systems, ICEIS 2020, Prague, Czech Republic, May*

5-7, 2020, Volume 1. SCITEPRESS, 2020, pp. 233–240. DOI: 10.5220/0009472702330240.

[10] Maria Krommyda, Verena Kantere, and Yannis Vassiliou. "Efficient Representation of Very Large Linked Datasets as Graphs". In: *Proceedings of the 22nd International Conference on Enterprise Information Systems, ICEIS 2020, Prague, Czech Republic, May 5-7, 2020, Volume 1*. SCITEPRESS, 2020, pp. 106–115. DOI: 10.5220/0009389001060115. URL: https://doi.org/10.5220/0009389001060115.

[11] Maria Krommyda, Anastasios Rigos, Spyridon-Nektarios Bolierakis, Theodoros Theodoropoulos, Stefano Tamascelli, Luca Simeone, Evangelos Sdongos, and Angelos Amditis. "An Integrated Toolbox for the Engagement of Citizens in the Monitoring of Water Ecosystems". In: *Electronics* 9.4 (2020), p. 671.

[12] Maria Krommyda, Anastatios Rigos, Kostas Bouklas, and Angelos Amditis. "Emotion detection in Twitter posts: a rule-based algorithm for annotated data acquisition". In: *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE. 2020.

[13] Anastasios Rigos, Maria Krommyda, Athanasia Tsertou, and Angelos Amditis. "A polynomial neural network for river's water-level prediction". In: *SN Applied Sciences* 2.4 (Mar. 2, 2020), p. 529. ISSN: 2523-3971. DOI: 10.1007/s42452-020-2328-9. URL: https://doi.org/10.1007/s42452-020-2328-9.

[14] Valantis Tsiakos, Maria Krommyda, Athanasia Tsertou, and Angelos Amditis. "From crowdsourcing environmental measurements to their integration in the GEOSS portal". In: *EGU General Assembly Conference Abstracts*. 2020, p. 21328.

[15] K. Tsitseklis, M. Krommyda, V. Karyotis, V. Kantere, and S. Papavassiliou. "Scalable Community Detection for Complex Data Graphs via Hyperbolic Network Embedding and Graph Databases". In: *IEEE Transactions on Network Science and Engineering* (2020), pp. 1–1. DOI: 10.1109/TNSE.2020.3022248.

[16] T. H. Assumpção, A. Jonoski, I. Theona, C. Tsiakos, M. Krommyda, S. Tamascelli, A. Kallioras, M. Mierla, H. V. Georgiou, M. Miska, C. Pouliaris, C. Trifanov, K. T. Cîmpan, A. Tsertou, E. Marin, M. Diakakis, I. Nichersu, A. J. Amditis, and I. Popescu. "Citizens' Campaigns for Environmental Water Monitoring: Lessons From Field Experiments". In: *IEEE Access* 7 (2019), pp. 134601–134620.

[17] M Krommyda, V Tsiakos, A Rigos, A Tsertou, A Amditis, H Georgiou, A Jonoski, I Popescu, and T Assumpcao. "Innovative sensors for crowdsourced river measurements collection". In: *CEST* (2019).

[18] Maria Krommyda and Verena Kantere. "Improving the Quality of the Conversational Datasets through Extensive Semantic Analysis". In: *2019 IEEE International Conference on Conversational Data & Knowledge Engineering (CDKE)*. IEEE. 2019, pp. 1–8.

[19] Maria Krommyda and Verena Kantere. "Understanding SPARQL endpoints through targeted exploration and visualization". In: *2019 First International Conference on Graph Computing (GC)*. IEEE. 2019, pp. 21–28.

[20] Maria Krommyda, Verena Kantere, and Yannis Vassiliou. "IVLG: Interactive Visualization of Large Graphs". In: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE. 2019, pp. 1984–1987.

[21] Maria Krommyda, Theodoros Theodoropoulos, Evangelos Sdongos, and Angelos Amditis. "Integrated monitoring system for environmental and river data measurements". In: *2019 13th International Conference on Sensing Technology (ICST)*. IEEE. 2019, pp. 1–6.

[22] C Tsiakos, M Krommyda, Y Kopsinis, A Tsertou, A Amditis, A Jonoski, I Popescu, and T Assumpcao. "Improved LC/LU maps and flood models through crowdsourced information". In: *CEST* (2019).

[23] Andreas Kallioras, Christos Pouliaris, Angelos Amditis, Athanasia Tsertou, Maria Krommyda, and Michail Diakakis. "Hydrologic monitoring supported by citizen sourced data for urban flood management and risk assessment". In: *AGUFM* 2018 (2018), H51X–1671.

[24] MARIA KROMMYDA, SPYROS BOLIERAKIS, YANNIS KOPSINIS, CHRYSOVALANTIS TSIAKOS, ATHANASIA TSERTOU, ANGELOS AMDITIS, ANDREJA JONOSKI, IOANA POPESCU, DANIELE MIORANDI, STEFANO TAMASCELLI, and BENJAMIN COHEN. "SCENT Integrated Toolbox for Monitoring Flood Phenomena". In: *WIT Transactions on The Built Environment* 184 (2018), pp. 121–132.

[25] Maria Krommyda, Evangelos Sdongos, Stefano Tamascelli, Athanasia Tsertou, Geli Latsa, and Angelos Amditis. "Towards citizen-powered cyberworlds for environmental monitoring". In: *2018 International Conference on Cyberworlds (CW)*. IEEE. 2018, pp. 454–457.

[26] Maria Krommyda, Athanasia Tsertou, Aggelos Amditis, Andreja Jonoski, Daniele Miorandi, and Benjamin Cohen. "Automated water lever and water surface velocity calculation from multimedia". In: *EGUGA* (2018), p. 14733.

[27] Luca Simeone, Silvia Brandalesi, Stefano Tamascelli, C Tsiakos, Maria Krommyda, Athanasia Tsertou, Angelos Amditis, and Amy Hume. "Best practices in serious games: Gamification strategies to support public participation in citizen observatories". In: *2nd International Conference Citizen Observatories for natural hazards and Water Management*. 2018.

[28] Nikos Bikakis, John Liagouris, Maria Krommyda, George Papastefanatos, and Timos Sellis. "GraphVizdb: A scalable platform for interactive large graph visualization". In: *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE. 2016, pp. 1342–1345.

[29] Nikos Bikakis, John Liagouris, Maria Kromida, George Papastefanatos, and Timos Sellis. "Towards scalable visual exploration of very large RDF graphs". In: *European Semantic Web Conference*. Springer. 2015, pp. 9–13.

Curriculum Vitae

I received my Diploma degree in Electrical and Computer Engineering from the National Technical University of Athens (NTUA) with specialization to computer system architecture and data management in 2013. Starting from January 2014 I began my PhD research focusing on the field of Big Data. Initially, focus was given to the visualization of big linked datasets bu then closely associated fields such as analytics, indexing, storage and management were investigated. In addition, I have gained experience as a teaching assistant and lab coordinator to the undergraduate Database management course. I am working, as a senior software developer, from June 2017, at ICCS/i-SENSE research institute focusing on Smart Integrated Systems and conducting scientific research and software development for the multiple EU funded projects. My main interests include among others: Architectural Design, User and System Requirements Definition, Database management, Data Indexing and Visualization, Big Data, Machine Learning, Computer Vision and Spatial Data Storage and Querying.