ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΣΥΣΤΗΜΑΤΩΝ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ ΚΑΙ ΜΑΘΗΣΗΣ

PAKTON: A Multi-Agent Framework for Question Answering in Long Legal Agreements

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Πέτρου Ραπτόπουλου

Επιβλέπων: Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2025

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης

PAKTON: A Multi-Agent Framework for Question Answering in Long Legal Agreements

Διπλωματική Εργασία

του

Πέτρου Ραπτόπουλου

Επιβλέπων: Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 3η Ιουλίου, 2025.

....
Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

....
Αθανάσιος Βουλόδημος
Επίκουρος Καθηγητής Ε.Μ.Π.

....
Ανδρέας-Γεώργιος Σταφυλοπάτης
Ομότιμος Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2025

....
Πετρος Ραπτοπουλος
Διπλωματούχος Ηλεκτρολόγος Μηχανικός
και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Πέτρος Ραπτόπουλος, 2025.
All rights reserved. Με επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Chapter 1

Περίληψη

Η ανασκόπηση συμβολαίων αποτελεί μια σύνθετη και χρονοβόρα διαδικασία, η οποία απαιτεί εξειδικευμένη νομική γνώση και, ως εκ τούτου, είναι σε μεγάλο βαθμό απρόσιτη σε μη ειδικούς. Επιπλέον, η νομική ερμηνεία σπάνια είναι απλή, η ασάφεια είναι συχνά παρούσα και οι αποφάσεις υπόκεινται σε υποκειμενικές ερμηνείες. Επιπροσθέτως, τα συμβόλαια είναι συνήθως εμπιστευτικά, γεγονός που περιορίζει τη χρήση τους με ιδιόκτητα μοντέλα και καθιστά απαραίτητη την προσφυγή σε ανοιχτού κώδικα εναλλακτικές λύσεις.

Για να αντιμετωπιστούν αυτές οι προκλήσεις, η παρούσα διπλωματική εργασία επεξηγεί και επεκτείνει το **PAKTON**[63]: ένα πλήρως ανοιχτού κώδικα, από άκρη σε άκρη, πολυπρακτορικό πλαίσιο με δυνατότητες εύκολης ενσωμάτωσης σε ήδη υπάρχοντα συστήματα.

Το προτεινόμενο σύστημα αποτελείται από τρεις πράκτορες: (1) τον *Αρχειοφύλακα*, ο οποίος αλληλεπιδρά με τον χρήστη και διαχειρίζεται την δομημένη αρχειόθετηση εγγράφων· (2) τον *Ερευνητή*, ο οποίος ανακτά πληροφορίες μέσω μιας καινοτόμου μονάδας Επαυξημένης Δημιουργίας μέσω Ανάκτησης (ΕΔΑ)· και (3) τον *Ανακριτή*, ο οποίος εμπλέκεται σε μια αλλεπάλληλη ανάκριση με στόχο τη σταδιακή βελτίωση της τελικής αναφοράς. Κάθε πράκτορας είναι προσαρμοσμένος στον ρόλο του, και μέσα από τη συνεργασία τους, παράγουν ένα ολοκληρωμένο και συνεκτικό αποτέλεσμα για τον τελικό χρήστη.

Η αρχιτεκτονική του PAKTON αποκλίνει από τα αδιαφανή μοντέλα, αποκαλύπτοντας ρητά τη συλλογιστική του διαδικασία και επισημαίνοντας τα αποσπάσματα τεκμηρίωσης — τόσο εντός του συμβολαίου του χρήστη όσο και από εξωτερικές νομικές πηγές — που οδήγησαν στις τελικές αποφάσεις. Το σύστημα αξιοποιεί προηγμένες τεχνικές ανάκτησης, όπως υβριδική πυκνή/αραιή ανάκτηση, γραφοευαίσθητη αναταξινόμηση και αποσπασματοποίηση βάσει συμφραζομένων, για την ανάδειξη των πιο σχετικών αποδεικτικών στοιχείων σε σχέση με το ερώτημα του χρήστη. Ο Ανακριτής λειτουργεί μέσω μια συνεχόμενης και επαναληπτικής βελτίωσης, εντοπίζοντας συστηματικά κενά στη λογική και ενισχύοντας την απάντηση σε κάθε επανάληψη, οδηγώντας έτσι σε μια πλήρη και πολυδιάστατη αναφορά.

Οι αξιολογήσεις σε πολλαπλά σύνολα δεδομένων, όπως τα ContractNLI και LegalBench-RAG, δείχνουν ότι το PAKTON υπερέχει σταθερά σε σχέση με γενικού σκοπού και εξειδικευμένα (σε νομικό τομέα) μοντέλα, τόσο σε ποσοτικούς δείκτες (π.χ. ακρίβεια, ανάκληση, ακρίβεια πρόβλεψης) όσο και σε ποιοτικές διαστάσεις (π.χ. επεξηγησιμότητα, συλλογιστική, πληρότητα). Οι αξιολογήσεις από ανθρώπους και από ΜΓΜ επιβεβαιώνουν την ανωτερότητα του PAKTON σε κρίσιμες πτυχές της νομικής τεχνητής νοημοσύνης, όπως η διαφάνεια, η διαχείριση αμφισημιών και η τεκμηρίωση, στοιχεία απαραίτητα για την υποστήριξη της λήψης αποφάσεων από τον τελικό χρήστη. Με τη καινοτόμα αρχιτεκτονική και τα αποτελέσματα της αξιολόγησης, το PAKTON αποτελεί σημαντική πρόοδο στον τομέα της νομικής τεχνητής νοημοσύνης, αποδεικνύοντας πως ο συντονισμός πολλών πρακτόρων και η στοχευμένη ανάκτηση μπορούν να ενισχύσουν ουσιαστικά τη συλλογιστική σε υψηλού ρίσκου πεδία.[1]

Λέξεις-κλειδιά — Συμβόλαια, νομική τεχνητή νοημοσύνη, Μεγάλα Γλωσσικά Μοντέλα, Επαυξημένη Δημιουργία μέσω Ανάκτησης, πολυπρακτορικά συστήματα, ερμηνευσιμότητα, διαφάνεια.

---

[1] Ο πηγαίος κώδικας είναι διαθέσιμος στο github.com/petrosrapto/PAKTON.

Chapter 2

Abstract

Contract review is a complex and time-intensive task that typically demands specialized legal expertise, rendering it largely inaccessible to non-experts. Moreover, legal interpretation is rarely straightforward—ambiguity is pervasive, and judgments often hinge on subjective assessments. Compounding these challenges, contracts are usually confidential, restricting their use with proprietary models and necessitating reliance on open-source alternatives.

To address these challenges, this thesis expands upon **PAKTON** [63]: a fully open-source, end-to-end, multi-agent framework with plug-and-play capabilities, designed to handle the complexities of contract analysis through collaborative agent workflows and a novel retrieval-augmented generation (RAG) component, enabling automated legal document review that is more accessible and privacy-preserving.

The proposed system is composed of three agents: (1) the *Archivist*, which interacts with the user and manages structured document input; (2) the *Researcher*, which retrieves relevant internal and external information using hybrid and graph-aware retrieval techniques; and (3) the *Interrogator*, which engages in multi-step reasoning to iteratively refine the final report. Each agent is tailored to its specific role, and through collaboration, they collectively produce a comprehensive outcome for the end user.

PAKTON's architecture departs from black-box LLM paradigms by explicitly exposing its reasoning process and highlighting the evidence spans—both within the user-provided contract and from external legal knowledge—that inform its conclusions. It leverages advanced retrieval techniques, including hybrid dense–sparse retrieval, graph-aware reranking, and context-sensitive chunking, to surface the most relevant evidence in response to user queries. The Interrogator agent operates through a multi-step refinement loop, systematically identifying reasoning gaps and enhancing the response with each iteration, resulting in a comprehensive report that reflects multiple perspectives.

Evaluations across multiple benchmarks, including ContractNLI and LegalBench-RAG, demonstrate that PAKTON consistently outperforms both general-purpose and legal domain–fine-tuned models in quantitative metrics (e.g., accuracy, recall, precision) and qualitative dimensions (e.g., explainability, reasoning, completeness). Human and LLM-based evaluations further confirm PAKTON's superiority in critical legal AI aspects such as transparency, ambiguity resolution, and evidentiary grounding, factors essential for supporting end-user decision-making. The framework is model-agnostic and suitable for deployment under privacy-preserving conditions, enhancing its practicality for real-world legal applications. With its modular architecture and rigorous benchmarking, PAKTON represents a significant step forward in legal AI, illustrating how multi-agent coordination and strategic retrieval can substantially elevate reasoning in high-stakes domains.[1]

Keywords — Contracts, legal AI, Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), explainable AI, multi-agent systems, few-shot prompting, document chunking, legal reasoning, LegalBench, ContractNLI, document retrieval, interpretability, transparency.

---

[1] Code available at github.com/petrosrapto/PAKTON.

Ευχαριστίες

Η ολοκλήρωση της παρούσας διπλωματικής εργασίας θα ήταν αδύνατη χωρίς την πολύπλευρη συμβολή και υποστήριξη ορισμένων εξαιρετικά σημαντικών ανθρώπων, στους οποίους οφείλω την ειλικρινή μου ευγνωμοσύνη.

Πρώτιστα, επιθυμώ να εκφράσω τις εγκάρδιες ευχαριστίες μου στον επιβλέποντα καθηγητή μου, κ. Στάμου Γεώργιο, για την τιμή και την ευκαιρία που μου προσέφερε να εργαστώ στο πλαίσιο του Εργαστηρίου Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης. Η επιστημονική καθοδήγηση και η διαρκής του ενθάρρυνση αποτέλεσαν πολύτιμο κίνητρο καθ' όλη τη διάρκεια της ακαδημαϊκής και προσωπικής πορείας. Η καθοδήγησή του δεν περιορίστηκε στο ακαδημαϊκό επίπεδο, αλλά επηρέασε ουσιαστικά και την επαγγελματική μου πορεία, ωθώντας με να εφαρμόσω στην πράξη τις θεωρητικές μου γνώσεις, να δοκιμαστώ σε πραγματικές εργασιακές συνθήκες και να αναπτύξω μια πολυδιάστατη και ισορροπημένη επαγγελματική ταυτότητα.

Θερμές ευχαριστίες οφείλω, επίσης, στους διδάκτορες κ. Γιώργο Φιλανδριανό και κα. Μαρία Λυμπεραίου για τη συνεχή και ουσιαστική τους στήριξη. Η αφοσίωσή τους, η διάθεσή τους για συνεργασία και η βαθιά επιστημονική τους κατάρτιση υπήρξαν ανεκτίμητες αρωγοί στην εξέλιξη της παρούσας εργασίας. Μέσω της ενεργού εμπλοκής τους, εντάχθηκα σε ένα δημιουργικό και δυναμικό περιβάλλον, από το οποίο αποκόμισα πολύτιμες εμπειρίες και δεξιότητες.

Τέλος, οφείλω ένα τεράστιο ευχαριστώ στους κοντινούς μου ανθρώπους. Οι γονείς μου, ο αδερφός μου και η κοπέλα μου, στάθηκαν δίπλα μου στις πιο δύσκολες στιγμές, σε δοκιμασίες υγείας που δεν θα μπορούσα να ανακάμψω μόνος.

Κατά τη διάρκεια όλης αυτής της διαδρομής, η υπενθύμιση ότι 'η πίεση είναι προσωρινή αλλά οι νίκες μένουν για πάντα' μου έδινε κουράγιο. Αυτή η διπλωματική εργασία, σηματοδοτώντας την ολοκλήρωση των σπουδών μου ενάντια στις αντιξοότητες, αποτελεί μια μεγάλη νίκη.

Πέτρος Ραπτόπουλος, Ιούλιος 2025

Contents

1 Περίληψη 5

2 Abstract 7

Contents 11

List of Figures 14

List of Tables 15

3 Εκτεταμένη Περίληψη στα Ελληνικά 19
- 3.1 Θεωρητικό Υπόβαθρο . . . 20
  - 3.1.1 Γλωσσικά Μοντέλα . . . 20
  - 3.1.2 Τεχνικές Μάθησης . . . 20
  - 3.1.3 Εκπαίδευση των Μεγάλων Γλωσσικών Μοντέλων . . . 21
  - 3.1.4 Η Έννοια της Προτροπής (Prompting) . . . 22
  - 3.1.5 Στρατηγικές και Τεχνικές Προτροπής . . . 23
  - 3.1.6 Επαυξημένη Δημιουργία μέσω Ανάκτησης (Retrieval-Augmented Generation - RAG) . 24
  - 3.1.7 Πολυπρακτορικά Συστήματα (Multi-Agent Systems) . . . 25
- 3.2 Η αρχιτεκτονική του PAKTON . . . 26
  - 3.2.1 Επισκόπηση . . . 26
  - 3.2.2 Αρχειοφύλακας . . . 27
  - 3.2.3 Ανακριτής . . . 28
  - 3.2.4 Ερευνητής . . . 30
- 3.3 Πειράματα και Αποτελέσματα . . . 30
  - 3.3.1 Ποσοτικά Αποτελέσματα . . . 31
  - 3.3.2 Ποιοτικά Αποτελέσματα . . . 32
  - 3.3.3 Αξιολόγηση μέσω LLMs . . . 33
- 3.4 Συμπεράσματα και Μελλοντικές Επεκτάσεις . . . 34
- 3.5 Περιορισμοί και Ηθική . . . 34

4 Introduction 37

5 Background 39
- 5.1 Large Language Models (LLMs) . . . 41
  - 5.1.1 Background . . . 41
  - 5.1.2 Transformer . . . 41
  - 5.1.3 Deeper Exploration of LLM Components . . . 43
- 5.2 Training Techniques . . . 45
  - 5.2.1 Overview . . . 46
  - 5.2.2 Pre-Training and Fine-Tuning . . . 46
- 5.3 Prompting . . . 48
  - 5.3.1 Prompting Methodology . . . 48
  - 5.3.2 Prompt Categories . . . 49

5.3.3 Prompting Techniques . . . 50
5.4 Retrieval-Augmented Generation (RAG) . . . 52
5.4.1 Motivation . . . 52
5.4.2 Core Architecture . . . 52
5.4.3 Retrieval Strategies . . . 53
5.4.4 Advantages of RAG . . . 53
5.5 Multi-Agent Systems . . . 53
5.5.1 Definition and Motivation . . . 53
5.5.2 Key Characteristics of Agents in MAS . . . 53
5.5.3 Agent Roles in LLM-based Systems . . . 54
5.5.4 Coordination Mechanisms . . . 54
5.6 Related Work . . . 54
5.6.1 Overview . . . 54
5.6.2 Key Inspirations and differentiations . . . 55

6 PAKTON Framework 57
6.1 Overview . . . 57
6.2 Archivist . . . 58
6.3 Interrogator . . . 59
6.4 Researcher . . . 61
6.5 Prompt Engineering . . . 62

7 Experiments and Results 63
7.1 Quantitative Results . . . 63
7.1.1 Performance on a classification dataset . . . 63
7.1.2 Performance of RAG . . . 64
7.2 Qualitative Results . . . 65
7.3 Evaluation using LLMs . . . 66

8 Conclusion 69

9 Limitations and Ethical Considerations 71
9.1 Limitations . . . 71
9.2 Ethical Considerations . . . 72

10 Future Work 73

11 Bibliography 75

A Framework's Implementation Details 81
A.1 Prompts . . . 81

B Experiments on LegalBenchRAG 83
B.1 Baselines . . . 83
B.2 PAKTON Evaluation Strategy . . . 83
B.3 Discussion . . . 84

C Experiments on ContractNLI 89
C.1 Dataset Overview . . . 89
C.2 Dataset Subsets . . . 89
C.3 Prompting Strategies . . . 89
C.4 Experimental Setup . . . 90
C.5 Detailed Results . . . 90
C.6 Execution Details of PAKTON . . . 90

D Human Evaluation 97

E G-EVAL Experiments **99**

F Representative Output from PAKTON and Comparative Analysis with GPT (RAG) **103**

List of Figures

3.1.1 (a) Μηδενικά παραδείγματα, (b) Ένα παράδειγμα, (c) Λίγα παραδείγματα [11] 23
3.1.2 Παράδειγμα προτροπής με ενδιάμεσες σκέψεις (Chain-of-Thought Prompting) [84] 24
3.2.1 Ροή χρήσης του PAKTON: υποβολή νομικού ερωτήματος και παραγωγή αναλυτικής αναφοράς. 26
3.2.2 Επισκόπηση του προτεινόμενου πλαισίου PAKTON και των εσωτερικών του συνιστωσών. 27
3.2.3 Αναπαράσταση Δέντρου και Ροή Τεμαχισμού για Συμβολαιακά Έγγραφα 29
3.3.1 Συγκριτική ανάλυση PAKTON και GPT-4o. Το άνω διάγραμμα απεικονίζει τις ανθρώπινες προτιμήσεις βάσει εννέα κριτηρίων αξιολόγησης, συγκεντρωτικά για όλες τις ερωτήσεις. Το κάτω διάγραμμα παρουσιάζει τα αποτελέσματα G-EVAL για τα ίδια κριτήρια, συγκεντρωτικά για όλες τις εξόδους στο ContractNLI. 33

5.1.1 **The Transformer - model architecture**. The original Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder, shown in the left and right halves of figure respectively[75] 43
5.1.2 Self-Attention Mechanism 44
5.3.1 Prompting Typology [48] 50
5.3.2 (a) Zero-shot prompting, (b) One-shot prompting, (c) Few-shot prompting [11] 51
5.3.3 Chain-of-thought [84] 52

6.1.1 PAKTON user flow: legal query submission followed by comprehensive report generation. 57
6.1.2 An overview of the proposed PAKTON framework and its internal components. 58
6.2.1 Tree Representation and Chunking Workflow for Contract Documents 60

7.2.1 Comparison analysis of PAKTON and GPT-4o. Top plot presents human preferences across nine evaluation criteria aggregated for all questions. Bottom plot shows G-EVAL scores for the same criteria, aggregated across all ContractNLI outputs. 65

B.3.1 Precision and Recall values for different k across all datasets for all methods 88

D.0.1 Comparative analysis of PAKTON vs. GPT-4o based on human evaluator judgments across different criteria 98

E.0.1 Comparative analysis of PAKTON vs. GPT-4o based on the G-EVAL framework across different criteria. Distribution of the aggregated scores across all criteria. 101

F.0.1 The user interface (UI) of PAKTON employed during the human evaluation with study participants. 106

List of Tables

7.1 Performance comparison of PAKTON and other methods across models on the ContractNLI test set. The highest accuracy and F1[w] are shown in **bold**. 67
7.2 Precision and Recall @ $k \in \{1, 2, 4, 8, 16, 32, 64\}$ for four retrieval pipelines on five legal-text datasets. 68

B.1 Performance comparison on different datasets for Precision and Recall at various k values for PAKTON's Researcher and Archivist under configuration 1. 86
B.2 Performance comparison across different datasets in terms of Precision and Recall at various k values, using PAKTON's Researcher and Archivist under Configuration 2. 86
B.3 Performance comparison across different datasets in terms of Precision and Recall at various k values, using PAKTON's Researcher and Archivist components under Configuration 1, based on span-based calculation of Precision and Recall. 87
B.4 Recall @ k for perfect retrieval using ground-truth snippets. Note: Precision is 100% for all cases. 87
B.5 Average number of characters retrieved @ k for each dataset for PAKTON configuration 1. Comparison with average length of ground truth (answer) and document lengths. 87

C.1 Baseline performance of models across multiple evaluation runs on the ContractNLI test set. (Part 1 of 3) 91
C.2 Baseline performance of models across multiple evaluation runs on the ContractNLI test set (Part 2 of 3). 92
C.3 Baseline performance of models across multiple evaluation runs on the ContractNLI test set (Part 3 of 3). 93
C.4 Detailed baseline performance of models across multiple evaluation runs. Evaluations are conducted on isolated spans—subsections of the contract text directly relevant to the hypothesis—. (Part 1 of 3) 94
C.5 Detailed baseline performance of models across multiple evaluation runs. Evaluations are conducted on isolated spans—subsections of the contract text directly relevant to the hypothesis—. (Part 2 of 3) 95
C.6 Detailed baseline performance of models across multiple evaluation runs. Evaluations are conducted on isolated spans—subsections of the contract text directly relevant to the hypothesis—. (Part 3 of 3) 96

E.1 Comparison of PAKTON and GPT scores acquired across different evaluation criteria aggregated for all examined outputs. 99
E.2 Example illustrating score penalization of "Contextual and Legal Understanding" due to further research section of the report. 100
E.3 Instructions given to human annotators for each evaluation criterion used in the PAKTON vs. ChatGPT comparison. Similar instructions were given to the G-EVAL framework. 100

Chapter 3

Εκτεταμένη Περίληψη στα Ελληνικά

3.1 Θεωρητικό Υπόβαθρο

3.1.1 Γλωσσικά Μοντέλα

Τα **Γλωσσικά Μοντέλα** (ΓΜ) είναι υπολογιστικά συστήματα που στοχεύουν στην κατανόηση και παραγωγή φυσικής γλώσσας, προβλέποντας την πιθανότητα εμφάνισης λέξεων ή φράσεων με βάση τα συμφραζόμενα. Στις πρώιμες μορφές τους, όπως τα *n-gram* μοντέλα, η πρόβλεψη βασιζόταν σε σταθερό αριθμό προηγούμενων λέξεων και σε απλούς πιθανοτικούς κανόνες. Παρά την απλότητά τους, τα μοντέλα αυτά περιορίζονταν από την αδυναμία τους να γενικεύσουν σε σπάνιες ή μη παρατηρημένες ακολουθίες λέξεων.

Για να ξεπεραστούν αυτοί οι περιορισμοί, εισήχθησαν τα **νευρωνικά γλωσσικά μοντέλα**, αρχικά μέσω *δίκτυα πρόσθιας τροφοδότησης* (feedforward neural networks [86]) και στη συνέχεια μέσω αναδρομικών (RNNs [54] , LSTMs). Κοινό τους χαρακτηριστικό είναι η χρήση *ενσωματώσεων λέξεων* (word embeddings), οι οποίες προσφέρουν συνεχείς αναπαραστάσεις υψηλότερης σημασιολογικής πληροφορίας. Αυτή η προσέγγιση επιτρέπει μεγαλύτερη προσαρμοστικότητα και κατανόηση συνθετότερων γλωσσικών δομών.

Η σημαντικότερη όμως τεχνολογική τομή ήρθε με την εισαγωγή της αρχιτεκτονικής των **Μετασχηματιστών** (Transformers) [76], που βασίζεται στον μηχανισμό *προσοχής* (attention) και εγκαταλείπει πλήρως τις επαναληπτικές δομές. Οι Μετασχηματιστές μπορούν να επεξεργάζονται ολόκληρες ακολουθίες λέξεων ταυτόχρονα, μοντελοποιώντας μακροχρόνιες εξαρτήσεις και πολύπλοκες γλωσσικές σχέσεις με εξαιρετική αποτελεσματικότητα. Αυτή η ιδιότητα τους καθιστά ιδανικούς για πλήθος εφαρμογών στην Επεξεργασία Φυσικής Γλώσσας (ΕΦΓ).

Μεγάλα Γλωσσικά Μοντέλα (ΜΓΜ). Τα **Μεγάλα Γλωσσικά Μοντέλα** (ΜΓΜ) είναι εξελιγμένα μοντέλα που βασίζονται στην αρχιτεκτονική του Μετασχηματιστή και εκπαιδεύονται σε τεράστια σώματα κειμένου. Λόγω του μεγέθους τους και της ποικιλίας των δεδομένων εκπαίδευσης, διαθέτουν αξιοσημείωτη ικανότητα γενίκευσης και αντιμετώπισης πολύπλοκων εργασιών.

Υπάρχουν τρεις βασικές αρχιτεκτονικές παραλλαγές ΜΓΜ:

- **Μόνο Κωδικοποιητής (Encoder-only):** Χαρακτηριστικό παράδειγμα είναι το *BERT*[20], το οποίο σχεδιάστηκε για εργασίες κατανόησης, όπως ταξινόμηση προτάσεων, αναγνώριση οντοτήτων και αντιστοίχιση ερωτήματος-απάντησης. Ο κωδικοποιητής αναλύει πλήρως την είσοδο, χωρίς να παράγει νέα ακολουθία εξόδου.
- **Κωδικοποιητής-Αποκωδικοποιητής (Encoder–Decoder):** Χρησιμοποιείται σε εργασίες όπου η είσοδος και η έξοδος είναι διαφορετικές ακολουθίες, όπως η μηχανική μετάφραση. Τα μοντέλα *BART*[43] και *T5*[62] αξιοποιούν αυτή τη δομή και διαπρέπουν σε καθήκοντα επεξεργασίας και αναδόμησης γλώσσας.
- **Μόνο Αποκωδικοποιητής (Decoder-only):** Μοντέλα όπως το *GPT*[61] ανήκουν σε αυτή την κατηγορία. Εκπαιδεύονται με αυτοπαλίνδρομη μάθηση (autoregressive learning), προβλέποντας διαδοχικά την επόμενη λέξη, γεγονός που τα καθιστά ιδανικά για δημιουργία κειμένου, συνομιλιακές διεπαφές, και σύνθετη γενετική συλλογιστική.

Αυτές οι αρχιτεκτονικές αντανακλούν διαφορετικές σχεδιαστικές φιλοσοφίες και είναι κατάλληλες για διαφορετικά είδη εργασιών στην ΕΦΓ. Τα encoder-only μοντέλα υπερέχουν σε κατανόηση και ανάλυση, τα encoder-decoder σε μετασχηματισμούς και μετάφραση, ενώ τα decoder-only είναι κορυφαία για γενετικές εφαρμογές.

Καθώς τα ΜΓΜ συνεχίζουν να εξελίσσονται—με την αύξηση του αριθμού παραμέτρων, τη βελτίωση των στρατηγικών εκπαίδευσης και την ενσωμάτωση εργαλείων και μνήμης—αποτελούν πλέον το θεμέλιο για την ανάπτυξη προηγμένων συστημάτων τεχνητής νοημοσύνης, μεταξύ των οποίων και νομικά εξειδικευμένα συστήματα όπως το *PAKTON*.

3.1.2 Τεχνικές Μάθησης

Οι τεχνικές **Μηχανικής Μάθησης** (Machine Learning – ML) αποτελούν το θεμέλιο της σύγχρονης τεχνητής νοημοσύνης και διαφοροποιούνται κυρίως βάσει του τρόπου με τον οποίο αξιοποιούνται τα δεδομένα κατά την εκπαίδευση, αλλά και του βαθμού εποπτείας που παρέχεται. Οι βασικές κατηγορίες περιλαμβάνουν την *επιβλεπόμενη*, τη *μη επιβλεπόμενη* και την *ενισχυτική μάθηση*, ενώ υπάρχουν και υβριδικές μορφές, όπως η *ημιεπιβλεπόμενη* και η *αυτοεπιβλεπόμενη μάθηση*, που συνδυάζουν χαρακτηριστικά από τις κύριες προσεγγίσεις.

Επιβλεπόμενη Μάθηση (Supervised Learning). Σε αυτό το πλαίσιο, το μοντέλο εκπαιδεύεται με *επισημασμένα* δεδομένα, δηλαδή κάθε είσοδος συνοδεύεται από μια γνωστή και σωστή έξοδο (ετικέτα). Στόχος είναι η μάθηση ενός μοντέλου πρόβλεψης που να μπορεί να γενικεύει σωστά σε νέα, μη ορατά δεδομένα. Η επιβλεπόμενη μάθηση εφαρμόζεται ευρέως σε προβλήματα όπως η ταξινόμηση (π.χ., ανάλυση συναισθήματος, διάγνωση ασθενειών) και η παλινδρόμηση (π.χ., πρόβλεψη τιμών ή ποσοτήτων). Η αποτελεσματικότητα αυτής της μεθόδου εξαρτάται άμεσα από την ποιότητα και την ποσότητα των διαθέσιμων επισημασμένων δεδομένων.

Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning). Σε αντίθεση με την επιβλεπόμενη προσέγγιση, εδώ το μοντέλο καλείται να ανακαλύψει δομές ή μοτίβα μέσα σε *μη επισημασμένα* δεδομένα, χωρίς να γνωρίζει προκαθορισμένες εξόδους. Κλασικές εφαρμογές περιλαμβάνουν την ομαδοποίηση (clustering), τη μείωση διαστάσεων (dimensionality reduction) και την ανίχνευση ανωμαλιών (anomaly detection). Η μη επιβλεπόμενη μάθηση είναι ιδιαίτερα χρήσιμη σε περιπτώσεις όπου δεν υπάρχουν διαθέσιμες ετικέτες, όπως σε ανάλυση καταναλωτικής συμπεριφοράς ή χαρτογράφηση θεμάτων σε μεγάλα κείμενα.

Ημιεπιβλεπόμενη Μάθηση (Semi-supervised Learning). Αυτή η τεχνική αποτελεί ενδιάμεση λύση, συνδυάζοντας έναν μικρό αριθμό επισημασμένων δεδομένων με ένα μεγαλύτερο σύνολο μη επισημασμένων. Το μοντέλο αρχικά εκπαιδεύεται στα δεδομένα με ετικέτες και κατόπιν αξιοποιεί την πληροφορία από τα μη επισημασμένα, βελτιώνοντας τη γενίκευση και μειώνοντας την ανάγκη για εκτενή χειροκίνητη επισημείωση. Είναι ιδιαίτερα αποτελεσματική σε πεδία όπου η επισήμανση είναι δαπανηρή ή χρονοβόρα, όπως η βιοϊατρική ή η νομική ανάλυση. [87]

Αυτοεπιβλεπόμενη Μάθηση (Self-supervised Learning). Η αυτοεπιβλεπόμενη μάθηση αναδεικνύεται ως ιδιαίτερα ισχυρή τεχνική, ειδικά για την προεκπαίδευση μεγάλων γλωσσικών και οπτικών μοντέλων. Σε αυτήν, το ίδιο το σύστημα δημιουργεί εποπτεία μέσω *τεχνητών προκαθορισμένων εργασιών* (pretext tasks), όπως η πρόβλεψη της επόμενης λέξης σε ένα κείμενο ή η συμπλήρωση μιας φράσης. Παρόλο που βασίζεται σε μη επισημασμένα δεδομένα, καταφέρνει να μάθει πλούσιες και γενικεύσιμες αναπαραστάσεις. Χρησιμοποιείται ευρέως στην Επεξεργασία Φυσικής Γλώσσας (ΕΦΓ) (π.χ., BERT, GPT) και στην Υπολογιστική Όραση (π.χ., SimCLR, DINO).

Ενισχυτική Μάθηση (Reinforcement Learning). Η ενισχυτική μάθηση διαφοροποιείται ουσιαστικά από τις προηγούμενες μεθόδους, καθώς στηρίζεται σε πράκτορες (agents) που *αλληλεπιδρούν με ένα περιβάλλον* και μαθαίνουν μέσω ανταμοιβών ή ποινών. Ο πράκτορας επιχειρεί ενέργειες, παρατηρεί τις συνέπειες και προσαρμόζει τη στρατηγική του ώστε να μεγιστοποιήσει τη σωρευτική ανταμοιβή. Η ενισχυτική μάθηση είναι η βάση για πολλά επιτεύγματα στην Τεχνητή Νοημοσύνη, όπως η εκμάθηση παιχνιδιών (π.χ., AlphaGo) και η βελτιστοποίηση στρατηγικών σε δυναμικά ή αβέβαια περιβάλλοντα (π.χ., ρομποτική, αυτόνομη πλοήγηση, οικονομική ανάλυση).

Συνοψίζοντας, κάθε τεχνική μάθησης παρουσιάζει μοναδικά πλεονεκτήματα και περιορισμούς, και η επιλογή εξαρτάται από τη φύση του προβλήματος, την ποιότητα των διαθέσιμων δεδομένων και τους στόχους του συστήματος. Σε πολλά σύγχρονα συστήματα, οι παραπάνω προσεγγίσεις συνδυάζονται για την επίτευξη ανώτερης απόδοσης και γενίκευσης.

3.1.3 Εκπαίδευση των Μεγάλων Γλωσσικών Μοντέλων

Η διαδικασία εκπαίδευσης των Μεγάλων Γλωσσικών Μοντέλων (ΜΓΜ) βασίζεται σε μια προσέγγιση δύο σταδίων: την αρχική *προεκπαίδευση* (pre-training) και την επακόλουθη *εξειδίκευση* (fine-tuning). Οι δύο αυτές φάσεις συνεργάζονται για να προσδώσουν στο μοντέλο τόσο γενική κατανόηση της φυσικής γλώσσας όσο και ικανότητα προσαρμογής σε συγκεκριμένες εφαρμογές.

Προεκπαίδευση (Pre-training). Στην πρώτη φάση, το μοντέλο εκπαιδεύεται σε τεράστιες ποσότητες αδόμητου ή ημιδομημένου κειμένου, όπως βιβλία, άρθρα ή ιστότοπους, χωρίς την απαίτηση επισημασμένων δεδομένων. Κατά την προεκπαίδευση, υλοποιούνται προγραμματισμένες εργασίες μάθησης, όπως η πρόβλεψη της επόμενης λέξης ή η ανασυγκρότηση ενός αποσπασματικού κειμένου, που επιτρέπουν στο μοντέλο να κατανοήσει τη γραμματική, τη σύνταξη και τις σημασιολογικές συσχετίσεις της γλώσσας. Αυτό προσφέρει στο μοντέλο μια ισχυρή βάση για μεταφορά γνώσης (transfer learning), μειώνοντας την ανάγκη για μεγάλα επισημασμένα σύνολα στις επόμενες φάσεις.

Παραδείγματα τέτοιων μοντέλων περιλαμβάνουν το BERT, το οποίο προεκπαιδεύεται μέσω μάσκας λέξεων, και το GPT, που βασίζεται στην αυτοπαλίνδρομη πρόβλεψη επόμενων λέξεων.

Εξειδίκευση (Fine-tuning). Μετά την απόκτηση γενικών γλωσσικών ικανοτήτων, το προεκπαιδευμένο μοντέλο προσαρμόζεται σε συγκεκριμένες εργασίες ή τομείς μέσω της διαδικασίας του fine-tuning. Σε αυτή τη φάση, το μοντέλο εκπαιδεύεται με επισημασμένα δεδομένα μικρότερης κλίμακας, επιτρέποντάς του να εξειδικεύσει τη γνώση του και να εντοπίσει πιο λεπτομερή μοτίβα που σχετίζονται με τη συγκεκριμένη εργασία, όπως η κατηγοριοποίηση νομικών ρητρών, η εξαγωγή οντοτήτων ή η απάντηση σε ερωτήματα.

Η τεχνική αυτή είναι ιδιαίτερα αποδοτική όταν τα διαθέσιμα δεδομένα είναι περιορισμένα, καθώς το μοντέλο ξεκινά με ένα προηγμένο επίπεδο γνώσης, μειώνοντας τον κίνδυνο υπερπροσαρμογής και βελτιώνοντας τη γενίκευση σε άγνωστα παραδείγματα.

Τομεακά Εξειδικευμένο Pre-training. Σε περιβάλλοντα όπου η εργασία-στόχος είναι ιδιαίτερα εξειδικευμένη (π.χ., νομικά ή ιατρικά κείμενα), η απόδοση του μοντέλου μπορεί να ενισχυθεί περαιτέρω με ένα ενδιάμεσο στάδιο: την *τομεακή προεκπαίδευση*. Σε αυτήν, το μοντέλο, αφού έχει ήδη αποκτήσει γενικές γλωσσικές γνώσεις, επανεκπαιδεύεται σε κείμενα του συγκεκριμένου πεδίου πριν την εφαρμογή του fine-tuning. Για παράδειγμα, ένα μοντέλο που έχει αρχικά εκπαιδευτεί στη Wikipedia, μπορεί στη συνέχεια να προεκπαιδευτεί σε νομικά έγγραφα ή ιατρικές αναφορές.

Αυτό το επιπλέον βήμα επιτρέπει στο μοντέλο να κατανοήσει το ορολογικό και θεματικό πλαίσιο του πεδίου εφαρμογής και βελτιώνει τη σημασιολογική του ευαισθησία σε τομεακά φαινόμενα. Το αποτέλεσμα είναι μεγαλύτερη ακρίβεια, συνέπεια και αξιοπιστία στις εξειδικευμένες εφαρμογές.

Συνδυαστική Στρατηγική Εκπαίδευσης. Ο συνδυασμός γενικής προεκπαίδευσης, τομεακής προεκπαίδευσης και στοχευμένου fine-tuning συγκροτεί μια ευέλικτη και ισχυρή στρατηγική. Αυτή η προσέγγιση προσδίδει στα ΜΓΜ ικανότητα γενίκευσης σε ευρύ φάσμα εργασιών, ενώ ταυτόχρονα επιτυγχάνουν υψηλή ακρίβεια σε ειδικές περιπτώσεις. Είναι μια τεχνική ιδιαίτερα κατάλληλη για σενάρια με περιορισμένα δεδομένα, καθώς μειώνει δραστικά τις απαιτήσεις για επισημασμένες εισόδους, ενισχύοντας την αποδοτικότητα, την επεκτασιμότητα και την πρακτική εφαρμοσιμότητα των μοντέλων στην πράξη.

Στην πράξη, αυτό το τριπλό σχήμα εκπαίδευσης έχει αποτελέσει τη βάση για πολλές επιτυχείς εφαρμογές σε πεδία όπως η νομική τεχνολογία, η ιατρική διάγνωση, η χρηματοοικονομική ανάλυση και οι εικονικοί βοηθοί φυσικής γλώσσας.

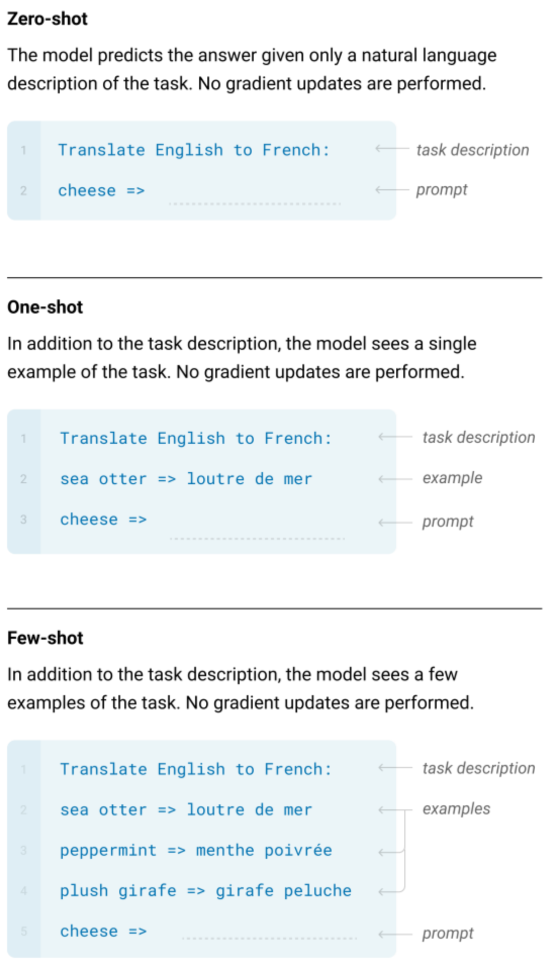
3.1.4 Η Έννοια της Προτροπής (Prompting)

- **Τυχαία επιλογή**: Ενδεικτικά παραδείγματα επιλέγονται χωρίς περαιτέρω φιλτράρισμα.
- **Σημασιολογική εγγύτητα**: Επιλέγονται παραδείγματα που είναι παρόμοια με τη νέα είσοδο, αυξάνοντας την συνάφεια.
- **Ποικιλομορφία παραδειγμάτων**: Η συμπερίληψη ετερόκλητων περιπτώσεων προωθεί καλύτερη γενίκευση.

Η **προτροπή** (prompting) αποτελεί μια θεμελιώδη καινοτομία στη χρήση των Μεγάλων Γλωσσικών Μοντέλων (ΜΓΜ), επαναπροσδιορίζοντας τον τρόπο με τον οποίο επιτυγχάνεται η προσαρμογή ενός μοντέλου σε νέες εργασίες. Σε αντίθεση με τις παραδοσιακές μεθόδους της επιβλεπόμενης μάθησης, όπου απαιτείται η εκ νέου εκπαίδευση του μοντέλου με επισημασμένα δεδομένα για κάθε διαφορετική εργασία, η μάθηση μέσω προτροπών επιτρέπει την αξιοποίηση της ήδη αποκτηθείσας γνώσης ενός προεκπαιδευμένου ΜΓΜ.

Στον πυρήνα αυτής της προσέγγισης βρίσκεται η ιδέα ότι οι είσοδοι προς το μοντέλο μπορούν να διατυπωθούν ως *οδηγίες σε φυσική γλώσσα* που ενεργοποιούν την εσωτερική γνώση του μοντέλου. Οι προτροπές συχνά περιλαμβάνουν κενά, παραδείγματα ή επεξηγηματικές οδηγίες, με σκοπό να υποβάλουν στο μοντέλο την κατάλληλη «ερμηνεία» της εργασίας και να το καθοδηγήσουν προς την επιθυμητή έξοδο.

Η προτροπή έχει αναδειχθεί ως ιδιαίτερα αποτελεσματική για σενάρια *μηδενικών* ή *λίγων παραδειγμάτων* (zero-shot και few-shot learning), όπου το μοντέλο καλείται να λύσει νέες εργασίες χωρίς εκτεταμένη περαιτέρω εκπαίδευση. Αυτή η ικανότητα καθιστά τα ΜΓΜ εξαιρετικά ευέλικτα και εύκολα επεκτάσιμα σε διαφορετικά περιβάλλοντα, προσφέροντας πρακτικές λύσεις σε ένα ευρύ φάσμα εφαρμογών.



Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
Translate English to French:
cheese =>
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
Translate English to French:
sea otter => loutre de mer
cheese =>
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>
```

Figure 3.1.1: (a) Μηδενικά παραδείγματα, (b) Ένα παράδειγμα, (c) Λίγα παραδείγματα [11]

3.1.5 Στρατηγικές και Τεχνικές Προτροπής

Η τέχνη του prompting αφορά την επιλογή, σύνθεση και διατύπωση της κατάλληλης μορφής προτροπής ώστε να αναδεικνύεται πλήρως η λειτουργικότητα του μοντέλου. Πέρα από τη βασική διατύπωση ερωτημάτων, οι τεχνικές prompting διαμορφώνονται βάσει της ποσότητας και της φύσης των πληροφοριών που παρέχονται στο μοντέλο. Οι κυριότερες κατηγορίες περιλαμβάνουν:

Μηδενικά Παραδείγματα (Zero-shot Prompting)

Στην τεχνική *zero-shot prompting*, το μοντέλο καλείται να απαντήσει σε μια εργασία μόνο βάσει της φρασεολογίας της οδηγίας, χωρίς να του παρέχεται κανένα παράδειγμα. Η επιτυχία αυτής της τεχνικής προϋποθέτει ότι το μοντέλο έχει αποκτήσει επαρκή εννοιολογική κατανόηση κατά το στάδιο της προεκπαίδευσης. Παρόλο που δεν παρέχεται εκ των προτέρων «μάθηση» μέσω παραδειγμάτων, η απόδοση των ΜΓΜ παραμένει εντυπωσιακή, ειδικά σε γνωστές και γενικές εργασίες.

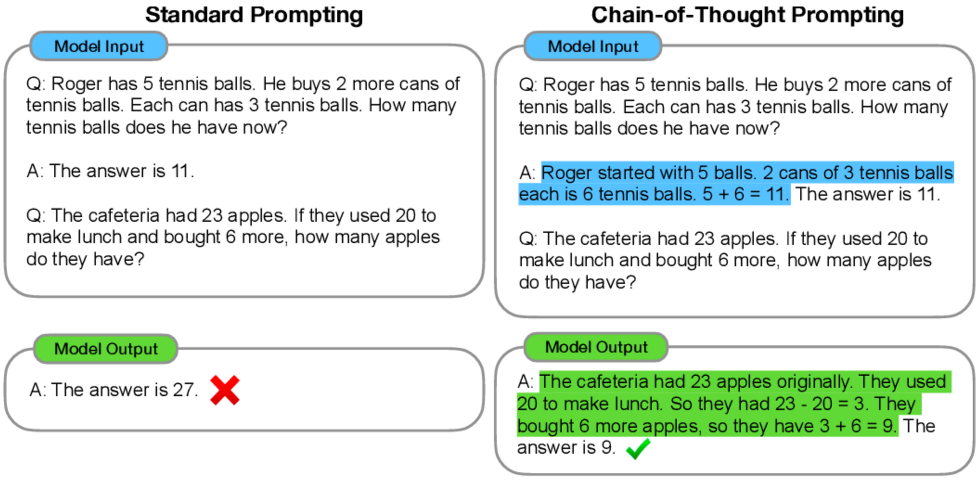
Ένα ή Λίγα Παραδείγματα (One-shot και Few-shot Prompting)

Στην περίπτωση της *one-shot prompting*, το μοντέλο λαμβάνει ένα μοναδικό παράδειγμα της ζητούμενης εργασίας, συνήθως με σαφή αντιστοίχιση εισόδου-εξόδου, και καλείται να επαναλάβει την απόδοση σε νέο δεδομένο. Η *few-shot prompting* επεκτείνει αυτή την ιδέα παρέχοντας στο μοντέλο πολλαπλά παραδείγματα (συνήθως 3–5), ενισχύοντας την ικανότητά του να αναγνωρίζει πρότυπα.

Η αποτελεσματικότητα αυτών των τεχνικών εξαρτάται από τον τρόπο επιλογής των παραδειγμάτων. Τα κυριότερα κριτήρια περιλαμβάνουν:

Προτροπή με Αλυσίδα Συλλογιστικής (Chain-of-Thought Prompting)

Η τεχνική *Chain-of-Thought Prompting* (CoT) εισάγει ένα κρίσιμο βήμα στην προτροπή, ενσωματώνοντας ρητά τη λογική πορεία επίλυσης ενός προβλήματος. Το μοντέλο ενθαρρύνεται να παραγάγει μια αλληλουχία συλλογισμών που οδηγούν στη λύση, προσομοιώνοντας τον τρόπο με τον οποίο ένας άνθρωπος σκέφτεται βήμα προς βήμα. Η προσέγγιση αυτή έχει αποδειχθεί ιδιαιτέρως αποτελεσματική σε εργασίες που απαιτούν μαθηματικό, αναλυτικό ή αιτιακό συλλογισμό, αυξάνοντας τη διαφάνεια και την αξιοπιστία των απαντήσεων.



Standard Prompting — Model Input:

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output:

A: The answer is 27.

Chain-of-Thought Prompting — Model Input:

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output:

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9.

Figure 3.1.2: Παράδειγμα προτροπής με ενδιάμεσες σκέψεις (Chain-of-Thought Prompting) [84]

Συνολικά, η στρατηγική του prompting αποτελεί έναν ιδιαίτερα αποδοτικό και ευέλικτο τρόπο αξιοποίησης των ΜΓΜ. Προσφέρει τη δυνατότητα δημιουργίας «plug-and-play» λύσεων, επιτρέποντας την εφαρμογή ενός μοντέλου σε νέα περιβάλλοντα χωρίς περαιτέρω επανεκπαίδευση. Οι τεχνικές prompting όχι μόνο επιταχύνουν τη διαδικασία προσαρμογής, αλλά ενισχύουν και τη διαφάνεια, καθιστώντας την προσέγγιση αυτή απαραίτητο εργαλείο στη σύγχρονη επεξεργασία φυσικής γλώσσας.

3.1.6 Επαυξημένη Δημιουργία μέσω Ανάκτησης (Retrieval-Augmented Generation - RAG)

Η Επαυξημένη Δημιουργία μέσω Ανάκτησης (**Retrieval-Augmented Generation – RAG**) αποτελεί ένα υβριδικό παράδειγμα συστήματος ερώτησης-απόκρισης, το οποίο συνδυάζει τη δύναμη της εξαγωγικής ανάκτησης πληροφοριών με την δημιουργική ικανότητα των Μεγάλων Γλωσσικών Μοντέλων (ΜΓΜ). Η κεντρική ιδέα πίσω από το RAG είναι η παροχή στο γλωσσικό μοντέλο πρόσβασης σε εξωτερικές πηγές γνώσης (π.χ., βάσεις εγγράφων, ιστοσελίδες, βάσεις δεδομένων), ώστε να μπορεί να δημιουργεί απαντήσεις που είναι όχι μόνο συντακτικά ορθές αλλά και επαληθεύσιμες και τεκμηριωμένες. Κατά αυτόν τον τρόπο αποφεύγονται φαινόμενα "παραισθήσεων" (hallucinations)[30].

Η παραδοσιακή προσέγγιση των ΜΓΜ βασίζεται αποκλειστικά στη γνώση που ενσωματώνεται κατά τη φάση της προεκπαίδευσης. Ωστόσο, η στατική φύση αυτής της γνώσης αποτελεί σημαντικό περιορισμό, ειδικά σε τομείς όπου οι πληροφορίες είναι δυναμικές ή εξειδικευμένες. Το RAG επιλύει αυτό το πρόβλημα επιτρέποντας στο μοντέλο να ανακτά σχετικά αποσπάσματα κειμένου από μια εξωτερική βάση δεδομένων (Retriever) και στη συνέχεια να δημιουργεί απαντήσεις βασισμένες σε αυτά τα αποσπάσματα (Generator).

Αυτή η αρχιτεκτονική αποτελείται από δύο κύρια στάδια:

- **Ανάκτηση (Retrieval):** Δίνεται ένα ερώτημα, και ο μηχανισμός ανάκτησης επιστρέφει τα πλέον σχετικά έγγραφα ή αποσπάσματα από μια συλλογή γνώσης.

- **Δημιουργία (Generation):** Το ΜΓΜ δημιουργεί την τελική απάντηση, ενσωματώνοντας το περιεχόμενο που ανακτήθηκε.

Η προσέγγιση RAG έχει αποδειχθεί ιδιαίτερα αποτελεσματική σε εφαρμογές όπως:

- Νομική και ιατρική τεκμηρίωση
- Εξειδικευμένη υποστήριξη πελατών
- Αναλυτική επεξεργασία επιστημονικών εγγράφων
- Εργαλεία συμβουλευτικής και διαλόγου με βάση τεκμήρια (evidence-based dialogue)

Η κύρια πρόκληση στα RAG συστήματα αφορά τη *ποιότητα της ανάκτησης*. Έστω και αν το γλωσσικό μοντέλο είναι ισχυρό, ελλιπής ή ανακριβής ανάκτηση μπορεί να οδηγήσει σε εσφαλμένες ή ανακριβείς απαντήσεις. Γι' αυτό, η βελτιστοποίηση τόσο του μηχανισμού ανάκτησης όσο και της στρατηγικής ολοκλήρωσης (fusion) παραμένει κρίσιμο ερευνητικό πεδίο.

3.1.7 Πολυπρακτορικά Συστήματα (Multi-Agent Systems)

Τα **πολυπρακτορικά συστήματα** (Multi-Agent Systems – MAS) αποτελούν ένα ισχυρό υπολογιστικό πλαίσιο για την επίλυση σύνθετων προβλημάτων μέσω της συνεργασίας πολλαπλών **πρακτόρων** (agents). Ένας πράκτορας μπορεί να οριστεί ως μια αυτόνομη υπολογιστική οντότητα με ικανότητα αντίληψης του περιβάλλοντός της, λήψης αποφάσεων και εκτέλεσης ενεργειών, με βάση συγκεκριμένους στόχους ή αποστολές. Οι πράκτορες διαθέτουν μηχανισμούς μάθησης, αντίδρασης και συχνά επικοινωνίας, καθιστώντας τους ιδανικούς για διαλογικά, αποκεντρωμένα ή συνεργατικά σενάρια.

Με την έλευση των Μεγάλων Γλωσσικών Μοντέλων (ΜΓΜ), η έννοια του πράκτορα εμπλουτίζεται: πλέον, ένας πράκτορας μπορεί να υλοποιηθεί ως ένα ΜΓΜ με *ενθυλακωμένη λογική απόφασης*, *πρόσβαση σε εργαλεία*, και *ικανότητα αυτόνομης αλληλεπίδρασης* με άλλα συστήματα ή χρήστες μέσω φυσικής γλώσσας. Για παράδειγμα, πράκτορες-ΜΓΜ μπορούν να συνδυάζουν διαλόγους, εργαλεία ανάκτησης γνώσης και σχεδιασμό ενέργειας, λειτουργώντας ως αυτόνομες υπομονάδες σε ένα μεγαλύτερο γνωσιακό σύστημα.

Ένα Πολυπρακτορικό Σύστημα (MAS) αποτελείται από πολλούς τέτοιους πράκτορες, που συνεργάζονται ή λειτουργούν παράλληλα εντός ενός κοινού περιβάλλοντος. Οι πράκτορες μπορεί να έχουν διαφορετική ειδίκευση ή στόχο, και η δυναμική αλληλεπίδραση μεταξύ τους μπορεί να επιφέρει λύσεις υψηλής πολυπλοκότητας ή να παράγει αναδυόμενες συμπεριφορές.

Κάθε πράκτορας σε ένα MAS μπορεί να διαθέτει:

- **Αυτόνομη λήψη αποφάσεων:** Εκτελεί ενέργειες βάσει εσωτερικής λογικής, στρατηγικής ή στόχων, χωρίς εξωτερικό έλεγχο.
- **Μερική γνώση:** Κατέχει μόνο ένα υποσύνολο της διαθέσιμης πληροφορίας, ενισχύοντας την ανάγκη συνεργασίας και συντονισμού.
- **Ικανότητα Επικοινωνίας:** Ανταλλάσσει πληροφορίες ή αιτήματα με άλλους πράκτορες μέσω κοινών πρωτοκόλλων επικοινωνίας, είτε κεντρικά είτε αποκεντρωμένα.

Η χρήση ΜΓΜ ως πράκτορες ενισχύει την ικανότητα κάθε μονάδας να κατανοεί σύνθετες γλωσσικές εντολές, να αναλύει έγγραφα, να αλληλεπιδρά διαλογικά με τον χρήστη και να χρησιμοποιεί εξωτερικά εργαλεία (όπως APIs, βάσεις δεδομένων, ή ανακτώμενη γνώση μέσω RAG). Παραδείγματα τέτοιων πρακτόρων περιλαμβάνουν:

- **Πράκτορες Ανάκτησης Γνώσης (Retrievers):** Εξειδικευμένοι στην εύρεση σχετικών πληροφοριών από εξωτερικές βάσεις.
- **Πράκτορες Συλλογισμού (Reasoners):** Παράγουν λογικές ακολουθίες βασισμένες σε ενδείξεις ή ανακτημένα αποσπάσματα.
- **Διαμεσολαβητές Συμπερασμάτων (Synthesizers):** Συνδυάζουν αποσπασματικές πληροφορίες σε ενιαίες, ερμηνεύσιμες απαντήσεις ή αναφορές.

Η κατανομή της γνωσιακής επιβάρυνσης μεταξύ των πρακτόρων οδηγεί σε μεγαλύτερη κλιμακωσιμότητα, εξειδίκευση και ανθεκτικότητα. Επιπλέον, επιτρέπει την επεξεργασία πολλαπλών στόχων ταυτόχρονα, την επανεξέταση υποθέσεων από διαφορετικές οπτικές γωνίες, και την παραγωγή συνεκτικών απαντήσεων μέσω διαλογικού ή ιεραρχικού συντονισμού μεταξύ των μοντέλων.

3.2 Η αρχιτεκτονική του PAKTON

Σε αυτήν την ενότητα, παρουσιάζουμε την πλήρη αρχιτεκτονική και τη λειτουργική ροή του **PAKTON**, του προτεινόμενου πολυπρακτορικού μας πλαισίου για την ανάλυση συμβολαίων. Η ενότητα περιγράφει πώς το PAKTON αντιμετωπίζει συστηματικά τις μοναδικές προκλήσεις της κατανόησης νομικών εγγράφων, διασπώντας το πρόβλημα σε διακριτά αλλά αλληλεπιδρώντα συστατικά μέρη. Ξεκινάμε με μια επισκόπηση υψηλού επιπέδου της λειτουργικότητας και των στόχων του συστήματος, εστιάζοντας στη διαφάνεια, τη θεμελίωση μέσω ανάκτησης πληροφορίας και την παραγωγή δομημένων νομικών αναφορών.

Στη συνέχεια, περιγράφουμε τους τρεις βασικούς πράκτορες—τον *Αρχειοφύλακα*, τον *Ανακριτή* και τον *Ερευνητή*—αναλύοντας τον ρόλο και την εσωτερική λογική καθενός στο συνολικό σύστημα επεξεργασίας. Ο *Αρχειοφύλακας* είναι υπεύθυνος για την εισαγωγή εγγράφων και τη ιεραρχική κατάτμησή τους, διευκολύνοντας την ανάκτηση πληροφορίας με γνώμονα το συμφραζόμενο. Ο *Ανακριτής* εκτελεί επαναληπτική λογική πάνω στα ανακτημένα αποσπάσματα, συνθέτοντας προσωποποιημένες νομικές αναφορές με ρητές αιτιολογήσεις και εντοπισμό κενών γνώσης. Ο *Ερευνητής* διαχειρίζεται την ενδο- και δια-εγγραφική ανάκτηση με υβριδικές και γραφοκεντρικές τεχνικές, ενισχυόμενος, όπου χρειάζεται, από εξωτερικά εργαλεία.

Κάθε υποενότητα εξηγεί τις τεχνικές λεπτομέρειες, τα κίνητρα σχεδίασης και τις αρχιτεκτονικές επιλογές που καθιστούν το **PAKTON** μία αποτελεσματική και εφαρμόσιμη λύση για την αυτοματοποιημένη ανασκόπηση συμβολαίων.

3.2.1 Επισκόπηση

Το **PAKTON** αποσκοπεί στην ανάλυση συμβολαίων που παρέχονται από τον χρήστη και στη δημιουργία απαντήσεων σε ερωτήματα, τεκμηριωμένων με βάση το περιεχόμενο του συμβολαίου, ενώ ενσωματώνει και σχετική εξωτερική γνώση, όπως απεικονίζεται στο Σχήμα 3.2.1. Κύριος στόχος του σχεδιασμού είναι η εξασφάλιση διαφανούς και ανιχνεύσιμης συλλογιστικής, μέσω αναφορών σε συγκεκριμένα αποσπάσματα του συμβολαίου και της σαφούς αιτιολόγησης κάθε συμπεράσματος. Η τελική έξοδος παρέχεται ως δομημένη νομική αναφορά.

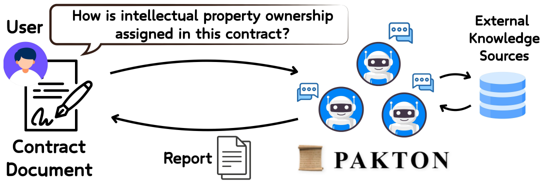

User
How is intellectual property ownership assigned in this contract?
External Knowledge Sources
Contract Document
Report
PAKTON

Figure 3.2.1: Ροή χρήσης του PAKTON: υποβολή νομικού ερωτήματος και παραγωγή αναλυτικής αναφοράς.

Για την υλοποίηση αυτής της λειτουργικότητας, το σύστημα βασίζεται σε μια αρχιτεκτονική τριών πρακτόρων—τον *Αρχειοφύλακα*, τον *Ανακριτή* και τον *Ερευνητή*—όπου καθένας επιτελεί έναν διακριτό ρόλο στην μονάδα ανάκτησης, λογισμού και σύνθεσης πληροφορίας. Κάθε πράκτορας υλοποιείται ως ένας βρόχος συλλογιστικής σύμφωνα με το πρότυπο *ReAct* [91], το οποίο συνδυάζει βήματα σκέψης και ενέργειας μέσω ενός κεντρικού Μεγάλου Γλωσικού Μοντέλου. Αυτός ο σχεδιασμός επιτρέπει στους πράκτορες να συλλογίζονται πάνω στην τρέχουσα κατάσταση, να παράγουν ενέργειες προσαρμοσμένες στο εργαλείο και να βελτιώνουν επαναληπτικά τις αποφάσεις τους βάσει ενδιάμεσων αποτελεσμάτων.

Μια συνολική απεικόνιση του πλαισίου παρουσιάζεται στο Σχήμα 3.2.2, ενώ επιπλέον τεχνικές λεπτομέρειες παρέχονται στο Παράρτημα A.

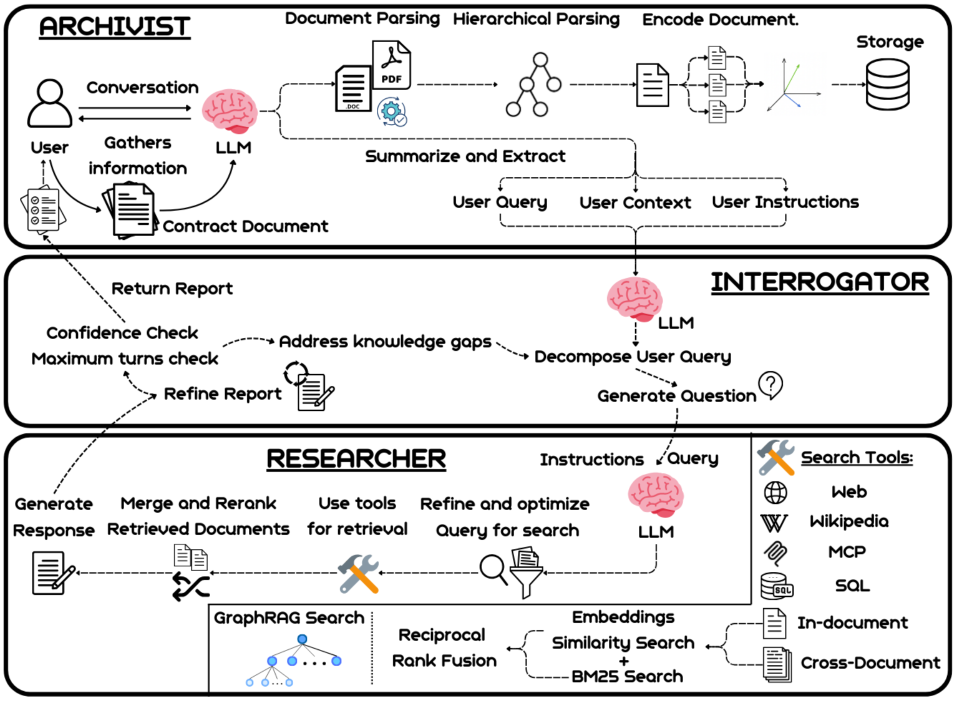


Figure 3.2.2: Επισκόπηση του προτεινόμενου πλαισίου PAKTON και των εσωτερικών του συνιστωσών.

3.2.2 Αρχειοφύλακας

Ο *Αρχειοφύλακας* είναι υπεύθυνος για τη συλλογή και οργάνωση των σχετικών πληροφοριών που παρέχει ο χρήστης. Διαδραματίζει κεντρικό ρόλο στα αρχικά στάδια του συστήματος, συγκεντρώνοντας και συνοψίζοντας το ερώτημα του χρήστη, τυχόν συνοδευτικές οδηγίες καθώς και σχετικό συμφραζόμενο υπόβαθρο. Οι πληροφορίες αυτές οργανώνονται και προωθούνται στον πράκτορα *Ανακριτή*. Ο *Αρχειοφύλακας* επίσης διαχειρίζεται την αποθήκευση και την ενσωμάτωση της διανυσματικής αναπαράστασης του συμβολαίου και αλληλεπιδρά ενεργά με τον χρήστη για την επίλυση ασαφειών ή την αναζήτηση ελλειπούς πληροφορίας. Εφαρμόζει τελικά τρεις βασικές λειτουργίες:

Ανάλυση Εγγράφου. Για να καταστεί δυνατή η πρακτική χρήση του συστήματος, ο *Αρχειοφύλακας* δέχεται διάφορους τύπους εγγράφων μέσω μιας ενοποιημένης ροής πληροφορίας. Αναλύει εγγενώς αρχεία κειμένου, διατηρώντας στοιχεία όπως τίτλους, παραγράφους και αριθμημένες διατάξεις. Για αρχεία PDF, εφαρμόζει τεχνικές OCR [71, 72] και ανάλυση οπτικής διάταξης για την αποκατάσταση της ιεραρχίας του εγγράφου, συμπεριλαμβανομένων ενοτήτων, πινάκων και διατάξεων πολλαπλών στηλών. Όλο το περιεχόμενο που εξάγεται φιλτράρεται και κανονικοποιείται ώστε να παρέχει συνεπή, δομημένη είσοδο για την επεξεργασία που ακολουθεί.

Ιεραρχική Ανάλυση. Τα νομικά έγγραφα παρουσιάζουν σύνθετες δομές, εσωτερικές παραπομπές και λεπτές σημασιολογικές διαφορές που καθιστούν τις επίπεδες αναπαραστάσεις ανεπαρκείς. Για την αντιμετώπιση αυτού, δημιουργούμε ένα ιεραρχικό δέντρο που αντανακλά τη δομή του ίδιου του συμβολαίου. Το κείμενο αναλύεται σε αυτή τη δομή με χρήση δομικών ενδείξεων σε συνδυασμό με μετρικές σημασιολογικής ομοιότητας βασισμένες σε ενσωματώσεις τύπου BERT [80] ή μέσω ΜΓΜ.

Η διαδικασία δομικής ανάλυσης περιλαμβάνει δύο κύρια βήματα, όπως φαίνεται στο Σχήμα 3.2.3: (1) αναγνώριση των διακριτών ενοτήτων του εγγράφου και (2) προσδιορισμός των ιεραρχικών σχέσεων μεταξύ τους. Παραδείγματα ενοτήτων περιλαμβάνουν τίτλους, ρήτρες, παραγράφους και αριθμημένα στοιχεία λίστας. Μία ενότητα θεωρείται «παιδί» κάποιας άλλης όταν εμφανίζεται ένθετη ως προς εκείνη, με βάση στοιχεία όπως η εσοχή, η αρίθμηση, η οπτική διάταξη (σε OCR έγγραφα) ή η σημασιολογική εγγύτητα.

Κωδικοποίηση Εγγράφου. Η αναπαράσταση του συμβολαίου ως δέντρο χρησιμοποιείται για τη δημιουργία αποσπασμάτων διατηρώντας τα συμφραζόμενα, τα οποία στη συνέχεια ενσωματώνονται για σκοπούς ανάκτησης πληροφορίας. Η τεμαχισμένη με βάση τα συμφραζόμενα αναπαράσταση έχει αποδειχθεί ότι μειώνει αποτυχίες ανάκτησης διατηρώντας τη σημασιολογική συνοχή μεταξύ ενοτήτων [4]. Ιδιαίτερα, τόσο η επικαλυπτόμενη [81] όσο και η δυναμική τεμαχισμένη αναπαράσταση [21] έχουν βελτιώσει τις επιδόσεις στην ανάκτηση πληροφορίας.

Για την κάλυψη διαφορετικών επιπέδων συμφραζομένων, δημιουργούμε τρεις τύπους αποσπασμάτων για κάθε κόμβο του δέντρου, τα οποία στη συνέχεια ενοποιούνται σε ένα ενιαίο σύνολο. Τα διπλότυπα ή σχεδόν όμοια αποσπάσματα φιλτράρονται ώστε να διατηρείται η ποικιλία και να αποφεύγεται η επαναληπτικότητα. Οι στρατηγικές τεμαχισμού είναι οι εξής:

- **Ανά Κόμβο:** Κωδικοποιεί κάθε κόμβο μεμονωμένα, διατηρώντας λεπτομερείς σημασιολογικές πληροφορίες. Ιδανικό για ακριβή αντιστοίχιση με ερωτήματα που στοχεύουν σε συγκεκριμένες ρήτρες ή νομικές διατάξεις.
- **Με Γνώση Προγόνων:** Συνδυάζει τον κόμβο με τους προγόνους του στην ιεραρχία, ενσωματώνοντας συμφραζόμενο από επικεφαλίδες και τίτλους. Αυτό βοηθά στην αποσαφήνιση και κατανόηση του ρόλου κάθε ενότητας στο γενικότερο πλαίσιο του συμβολαίου.
- **Με Γνώση Απογόνων:** Συνενώνει τον κόμβο με τους απογόνους του για την ενσωμάτωση συνεκτικών ενοτήτων, όπως ρήτρες και υπο-ρήτρες ή λίστες. Ενδείκνυται για συλλογιστική πάνω σε σύνθετες ή πολυμερείς διατάξεις.

Κάθε απόσπασμα εμπλουτίζεται με μεταδεδομένα, όπως η θέση του στον δομικό κορμό, η θέση του στο έγγραφο, το όνομα του αρχείου και μία συνοπτική περίληψη του συμβολαίου. Αυτά τα μεταδεδομένα διευκολύνουν τόσο την ανάκτηση μεταξύ εγγράφων (μέσω δρομολόγησης στην αποθήκη διανυσμάτων) όσο και την ενδο-εγγραφική κατάταξη συνάφειας.

Με την ενσωμάτωση πολλαπλών δομικών οπτικών, το πλαίσιο τεμαχισμού αυξάνει την πιθανότητα ανάκτησης σχετικής πληροφορίας για ένα ευρύ φάσμα ερωτημάτων—από αναζητήσεις ρητρών έως συλλογισμούς σε επίπεδο ενοτήτων.

3.2.3 Ανακριτής

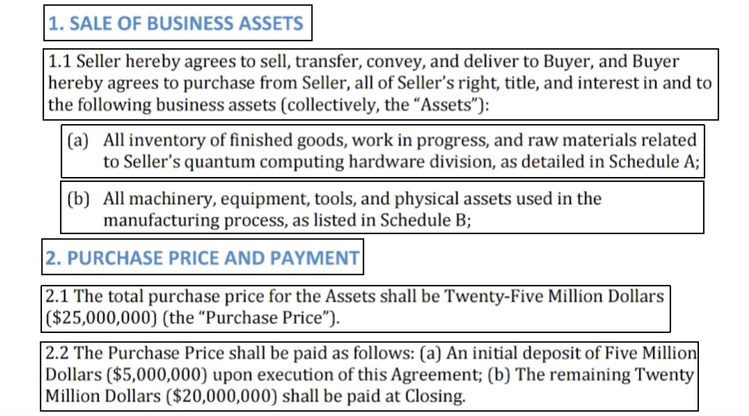
Ο *Ανακριτής* είναι υπεύθυνος για τη δημιουργία της τελικής αναφοράς που παρουσιάζεται στον χρήστη ως απάντηση του συστήματος. Για τον σκοπό αυτό, συντονίζει μια πολυσταδιακή διαδικασία συλλογισμού με στόχο την ακριβή και τεκμηριωμένη απάντηση στο ερώτημα. Ως είσοδο δέχεται το ερώτημα του χρήστη, το συμφραζόμενο πλαίσιο και τις σχετικές οδηγίες, και ξεκινά μια επαναληπτική «ανάκριση» του πράκτορα *Ερευνητή*. Η διαδικασία αυτή περιλαμβάνει τη διάσπαση του αρχικού ερωτήματος και τη δημιουργία σειράς υποερωτημάτων που αποσκοπούν στην εμβάθυνση και αποσαφήνιση της πληροφορίας.

Αφού λάβει απάντηση στο πρώτο υποερώτημα από τον *Ερευνητή*, ο *Ανακριτής* δημιουργεί ένα προκαταρκτικό προσχέδιο αναφοράς που επιχειρεί να απαντήσει άμεσα στο αρχικό ερώτημα του χρήστη. Η αρχική αυτή απόκριση αξιοποιείται επίσης για τον εντοπισμό πιθανών κενών γνώσης που απαιτούν περαιτέρω διερεύνηση.

Βάσει της αρχικής ανταλλαγής, ο *Ανακριτής* διαμορφώνει επακόλουθα ερωτήματα με στόχο την κάλυψη των κενών στην προκαταρκτική απάντηση. Κατά τη δημιουργία κάθε νέου ερωτήματος, λαμβάνει υπόψη το αρχικό ερώτημα, τα συμφραζόμενα, τις οδηγίες, μια σύνοψη των προηγούμενων ερωτημάτων και το τρέχον προσχέδιο της αναφοράς. Δίνεται προτεραιότητα σε ερωτήματα που αναμένεται να βελτιώσουν ουσιαστικά την ποιότητα της τελικής απάντησης.

Με κάθε νέα απάντηση από τον *Ερευνητή*, ο *Ανακριτής* βελτιώνει προοδευτικά το προσχέδιο της αναφοράς. Η διαδικασία συνεχίζεται μέχρι να ισχύσει ένα από τα δύο κριτήρια τερματισμού: (1) διαπιστώνεται ότι η απάντηση είναι επαρκώς πλήρης και έγκυρη ή (2) έχει επιτευχθεί το μέγιστο επιτρεπόμενο πλήθος επαναλήψεων ανάκρισης (που προσομοιώνει το βάθος ανάκτησης), όπως ορίζεται από τον χρήστη.

Ο *Ανακριτής* καθοδηγείται ρητά ώστε να αποφεύγει υποθέσεις και να παραμένει θεμελιωμένος στο ανακτημένο περιεχόμενο, επισημαίνοντας με σαφήνεια τυχόν ασάφειες ή ελλείψεις που εμποδίζουν την πλήρη απάντηση του ερωτήματος.



1. SALE OF BUSINESS ASSETS

1.1 Seller hereby agrees to sell, transfer, convey, and deliver to Buyer, and Buyer hereby agrees to purchase from Seller, all of Seller's right, title, and interest in and to the following business assets (collectively, the "Assets"):

(a) All inventory of finished goods, work in progress, and raw materials related to Seller's quantum computing hardware division, as detailed in Schedule A;

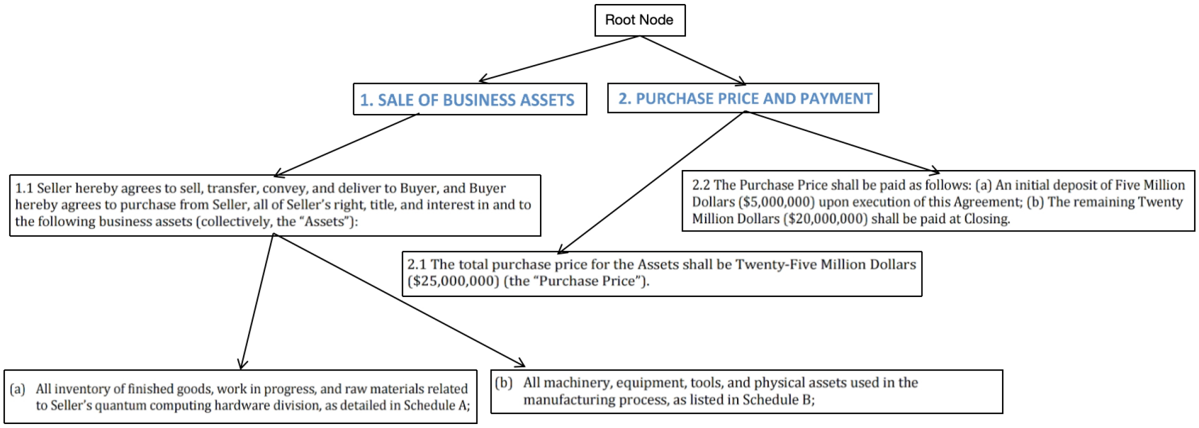
(b) All machinery, equipment, tools, and physical assets used in the manufacturing process, as listed in Schedule B;

2. PURCHASE PRICE AND PAYMENT

2.1 The total purchase price for the Assets shall be Twenty-Five Million Dollars ($25,000,000) (the "Purchase Price").

2.2 The Purchase Price shall be paid as follows: (a) An initial deposit of Five Million Dollars ($5,000,000) upon execution of this Agreement; (b) The remaining Twenty Million Dollars ($20,000,000) shall be paid at Closing.

(a) Εντοπισμός Ενοτήτων



1.1 Seller hereby agrees to sell, transfer, convey, and deliver to Buyer, and Buyer hereby agrees to purchase from Seller, all of Seller's right, title, and interest in and to the following business assets (collectively, the "Assets"):

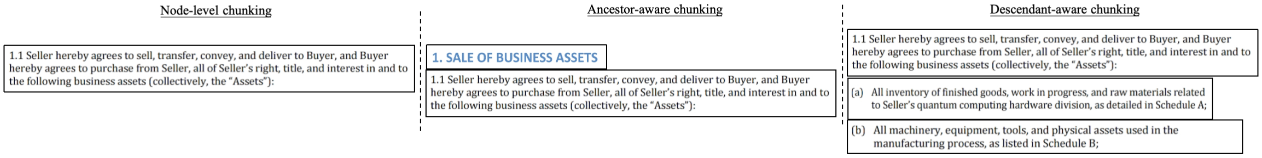
2.2 The Purchase Price shall be paid as follows: (a) An initial deposit of Five Million Dollars ($5,000,000) upon execution of this Agreement; (b) The remaining Twenty Million Dollars ($20,000,000) shall be paid at Closing.

2.1 The total purchase price for the Assets shall be Twenty-Five Million Dollars ($25,000,000) (the "Purchase Price").

(a) All inventory of finished goods, work in progress, and raw materials related to Seller's quantum computing hardware division, as detailed in Schedule A;

(b) All machinery, equipment, tools, and physical assets used in the manufacturing process, as listed in Schedule B;

(b) Ιεραρχική Δομή Ενοτήτων



1.1 Seller hereby agrees to sell, transfer, convey, and deliver to Buyer, and Buyer hereby agrees to purchase from Seller, all of Seller's right, title, and interest in and to the following business assets (collectively, the "Assets"):

1. SALE OF BUSINESS ASSETS

1.1 Seller hereby agrees to sell, transfer, convey, and deliver to Buyer, and Buyer hereby agrees to purchase from Seller, all of Seller's right, title, and interest in and to the following business assets (collectively, the "Assets"):

1.1 Seller hereby agrees to sell, transfer, convey, and deliver to Buyer, and Buyer hereby agrees to purchase from Seller, all of Seller's right, title, and interest in and to the following business assets (collectively, the "Assets"):

(a) All inventory of finished goods, work in progress, and raw materials related to Seller's quantum computing hardware division, as detailed in Schedule A;

(b) All machinery, equipment, tools, and physical assets used in the manufacturing process, as listed in Schedule B;

(c) Συμφραζόμενες Ενσωματώσεις για τον κόμβο "1.1 Seller ..."

Figure 3.2.3: Αναπαράσταση Δέντρου και Ροή Τεμαχισμού για Συμβολαιακά Έγγραφα

Επιπλέον, ο *Ανακριτής* είναι υπεύθυνος για την επικύρωση της δομής της τελικής αναφοράς, η οποία πρέπει να περιλαμβάνει:

- Τίτλο και συνοπτική Περίληψη Θέματος.
- Νομικό συλλογισμό και βασικά συμπεράσματα, για ενίσχυση της διαφάνειας
- Προκαταρκτική απάντηση και κατευθύνσεις για περαιτέρω διερεύνηση
- Εντοπισμένα κενά γνώσης και προτεινόμενα επόμενα ερωτήματα
- Λίστα υποστηρικτικών πηγών

3.2.4 Ερευνητής

Ο *Ερευνητής* είναι υπεύθυνος για την ανάκτηση σχετικής πληροφορίας προκειμένου να υποστηρίξει τον *Ανακριτή* στην απάντηση του ερωτήματος του χρήστη. Ως πράκτορας, είναι εξοπλισμένος με πολλαπλές μεθόδους ανάκτησης, καθεμία εκ των οποίων υλοποιείται ως ξεχωριστό εργαλείο. Ανάλογα με τη φύση του ερωτήματος και τις σχετικές οδηγίες, ο *Ερευνητής* επιλέγει αυτόνομα τη βέλτιστη μέθοδο ή συνδυασμό μεθόδων ανάκτησης. Η επιλογή αυτή καθοδηγείται μέσω προτροπής σε ΜΓΜ, το οποίο αξιολογεί το περιεχόμενο του ερωτήματος και τις περιγραφές των εργαλείων για να επιλέξει την κατάλληλη στρατηγική.

Οι μέθοδοι ανάκτησης κατηγοριοποιούνται σε δύο βασικούς τύπους:

- **Ανάκτηση εντός εγγράφου**: Αξιοποιώντας τα αποσπάσματα και τις διανυσματικές τους αναπαραστάσεις που παράγονται από τον *Αρχειοφύλακα*, ο *Ερευνητής* ανακτά σχετικές ενότητες για την απάντηση του ερωτήματος. Χρησιμοποιείται μια υβριδική προσέγγιση ανάκτησης που συνδυάζει BM25[64], πυκνά embeddings[44], και το Reciprocal Rank Fusion (RRF)[17]. Επιπλέον, ενσωματώνεται το *LightRAG*[31] (μια ελαφριά εκδοχή του *GraphRAG* [22]), το οποίο ενισχύει την αντιστοίχιση σε επίπεδο οντοτήτων και σχέσεων, εντός και μεταξύ εγγράφων. Αυτό είναι ιδιαίτερα αποτελεσματικό για ερωτήματα που απαιτούν λογισμό πάνω σε πολλαπλά αλληλεξαρτώμενα έγγραφα.
- **Ανάκτηση μεταξύ εγγράφων**: Αυτή η μέθοδος ανακτά σχετικές ενότητες από άλλα έγγραφα, τις οποίες ο *Ανακριτής* μπορεί να αξιοποιήσει ως παραδείγματα μέσω τεχνικών προτροπής παραδειγμάτων.

  Το PAKTON υποστηρίζει επίσης επιπλέον δυνατότητες ανάκτησης, όπως αναζήτηση στο διαδίκτυο, πρόσβαση στη Wikipedia, ερωτήματα σε βάσεις δεδομένων SQL και σύνδεση με εξωτερικές νομικές βάσεις μέσω του Model Context Protocol (MCP)[5]. Τα εργαλεία αυτά έχουν σχεδιαστεί ώστε να ενισχύουν τις γνώσεις του μοντέλου όταν αυτό δεν διαθέτει επαρκή πληροφόρηση ή απαιτείται ενημέρωση σε πραγματικό χρόνο. Ωστόσο, για λόγους δίκαιης και ελεγχόμενης αξιολόγησης, αυτή η κατηγορία ανάκτησης δεν χρησιμοποιήθηκε στα πειράματά μας.

Για την επίτευξη τόσο υψηλής ανάκλησης όσο και ακρίβειας, ο *Ερευνητής* εφαρμόζει μια διαδικασία δύο σταδίων: ανάκτηση και επανακατάταξη. Αρχικά, χρησιμοποιούνται ανακτήσεις με υψηλή ανάκληση (όπως αυτές που περιγράφηκαν παραπάνω) για να συλλεχθεί ένα ευρύ σύνολο υποψήφιων αποσπασμάτων με σκοπό τη μέγιστη κάλυψη. Στη συνέχεια, ένα μοντέλο cross-encoder επανακατατάσσει τα αποτελέσματα, κωδικοποιώντας από κοινού το ερώτημα και κάθε απόσπασμα, ώστε να εκτιμήσει με μεγαλύτερη ακρίβεια τη σημασιολογική συνάφεια[37]. Αυτή η φάση λειτουργεί ως φίλτρο υψηλής ακρίβειας, βελτιώνοντας το αρχικό σύνολο υποψηφίων. Οι βαθμολογίες συνάφειας κανονικοποιούνται μέσω μιας sigmoid συνάρτησης, και όσα αποσπάσματα υπερβαίνουν ένα προκαθορισμένο όριο περνούν στη φάση παραγωγής απάντησης.

3.3 Πειράματα και Αποτελέσματα

Σε αυτή την ενότητα παρουσιάζονται τα πειράματα και τα αποτελέσματα με σκοπό την αξιολόγηση της αποτελεσματικότητας του προτεινόμενου πλαισίου σε πολλαπλές διαστάσεις αξιολόγησης. Το **PAKTON** αξιολογείται τόσο με ποσοτικές όσο και με ποιοτικές μεθόδους. Στην ποσοτική ανάλυση, δοκιμάζεται σε ένα περιβάλλον ανάλυσης συμβολαίων, όπου υπερέχει όλων των βασικών συγκριτικών μοντέλων ως προς την ακρίβεια. Για την απομόνωση της συνεισφοράς μεμονωμένων υποσυστημάτων, οι ενότητες *Αρχειοφύλακα-Ερευνητή*, που υλοποιούν τον μηχανισμό Επαυξημένης Δημιουργίας μέσω Ανάκτησης (ΕΔΑ), αξιολογούνται σε τέσσερα σύνολα δεδομένων αναφοράς, επιτυγχάνοντας καινοτόμα αποτελέσματα.

Για τον *Ανακριτή*—ο οποίος είναι υπεύθυνος για τη δημιουργία της τελικής απάντησης προς τον χρήστη—διεξάγεται ποιοτική αξιολόγηση, κατά την οποία οι έξοδοι εξετάζονται ως προς εννέα κριτήρια που ευθυγραμμίζονται με τον στόχο του συστήματος να παράγει ερμηνεύσιμες και αξιοποιήσιμες απαντήσεις. Η αξιολόγηση αυτή περιλαμβάνει δομημένη ανθρώπινη κρίση από πέντε έμπειρους νομικούς και έναν Δικαστή του Αρείου Πάγου, καθώς και αυτόματη αξιολόγηση μέσω G-EVAL [49]. Σε αμφότερες τις αξιολογήσεις, το **PAKTON** ξεπερνά σταθερά το **GPT-4o**, υπερέχοντας σε οκτώ από τα εννέα κριτήρια.

3.3.1 Ποσοτικά Αποτελέσματα

Η ποσοτική αξιολόγηση παρέχει ένα αυστηρό και αντικειμενικό θεμέλιο για την εκτίμηση της αποτελεσματικότητας και της ανθεκτικότητας του πλαισίου **PAKTON**. Σε αυτή την ενότητα παρουσιάζουμε εμπειρικά αποτελέσματα από σύνολα δεδομένων αναφοράς που μετρούν ποσοτικά τόσο την ακρίβεια ταξινόμησης όσο και την απόδοση ανάκτησης. Οι μετρικές αυτές επιτρέπουν την απομόνωση και σύγκριση της συνεισφοράς επιμέρους υποσυστημάτων, όπως οι μονάδες συλλογιστικής και ανάκτησης, υπό ελεγχόμενες συνθήκες. Συγκρίνοντας με ισχυρές γραμμές βάσης—συμπεριλαμβανομένων ειδικά προεκπαιδευμένων νομικών μοντέλων και προηγμένων τεχνικών Επαυξημένης Δημιουργίας μέσω Ανάκτησης (ΕΔΑ)—δείχνουμε ότι το PAKTON όχι μόνο επιτυγχάνει ανώτερη προβλεπτική απόδοση, αλλά παρουσιάζει και αξιοσημείωτη ανθεκτικότητα ανεξαρτήτως του υποκείμενου γλωσσικού μοντέλου. Τα ευρήματα αυτά προσφέρουν σαφή απόδειξη της γενικευσιμότητας, της αποτελεσματικότητας και της πρακτικής χρησιμότητας του πλαισίου για πραγματικές εργασίες ανάλυσης συμβολαίων.

Απόδοση σε Δεδομένα Ταξινόμησης

Διαμόρφωση Πειράματος. Για την ποσοτική αξιολόγηση, το PAKTON δοκιμάστηκε στο σύνολο δεδομένων *ContractNLI* [38]. Σε αυτό, κάθε τεκμήριο (premise) αντιστοιχεί σε πλήρες κείμενο συμβολαίου, το οποίο συσχετίζεται με μία υπόθεση (hypothesis). Η εργασία ταξινόμησης έγκειται στον προσδιορισμό εάν η υπόθεση προκύπτει από, έρχεται σε αντίφαση με, ή είναι ουδέτερη ως προς το συμβόλαιο. Δεδομένου του μεγάλου μήκους των κειμένων, το σύνολο δεδομένων παρέχει σχολιασμένα αποσπάσματα που υποδεικνύουν τις κρίσιμες περιοχές για τη λήψη απόφασης.

Το PAKTON αξιολογείται συγκριτικά με πολλαπλές γραμμές βάσης, συμπεριλαμβανομένων μοντέλων προεκπαιδευμένων σε νομικά σώματα κειμένου (π.χ. Saul [15]) και διαφορετικών τεχνικών prompting (Πίνακας 7.1). Οι μετρικές περιλαμβάνουν συνολική ακρίβεια και σταθμισμένο F1 (F1[W]), καθώς και μεμονωμένα F1 για τις κατηγορίες: συνεπαγωγή (entailment), αντίφαση (contradiction) και ουδετερότητα (neutral).

Οι τεχνικές προτροπής περιλαμβάνουν zero-shot (ZS), few-shot (FS)—όπου παρέχονται πλήρη συμβόλαια ως παραδείγματα—και FS-Spans, όπου δίνονται μόνο τα κρίσιμα αποσπάσματα. Δείτε το Παράρτημα C για λεπτομέρειες του πειραματικού σχεδιασμού. Τα αποτελέσματα αναδεικνύουν σαφώς την υπεροχή του PAKTON, ξεπερνώντας ακόμα και ειδικά εκπαιδευμένα μοντέλα.

Αποτελέσματα. Το PAKTON υπερέχει σταθερά των γραμμών βάσης σε όλα τα μοντέλα. Συγκρίνοντας τα Mistral και Mixtral με το Saul, παρατηρείται ότι το PAKTON αποδίδει καλύτερα ακόμα και σε σχέση με fine-tuned μοντέλα επί νομικών δεδομένων. Ενδεικτικά, το Mistral 7B με PAKTON ξεπερνά το Saul 54B, παρά τη διαφορά μεγέθους.

Επιπλέον, η διακύμανση της απόδοσης μεταξύ διαφορετικών LLMs που ενσωματώνουν το PAKTON είναι μικρή, υποδεικνύοντας υψηλή ανθεκτικότητα. Για παράδειγμα, το F1[W] του Llama 3.1 70B με PAKTON είναι 79.03%, ενώ του Gemma 3 27B είναι 82.83% (διαφορά μόλις 3.8 μονάδες). Αντίθετα, στη ZS ρύθμιση η διαφορά μεταξύ των ίδιων μοντέλων αγγίζει τις 22.83 ποσοστιαίες μονάδες. Αυτό δείχνει ότι η αρχιτεκτονική του PAKTON μειώνει τη διακύμανση απόδοσης μεταξύ μοντέλων.

Για να εξετάσουμε περαιτέρω την ανθεκτικότητα, υπολογίζουμε τον συντελεστή μεταβλητότητας (CV) για το F1[W] σε όλα τα μοντέλα με PAKTON και βρίσκουμε $CV = 12.6\%$, έναντι > 25% για ZS. Επιπλέον, εκτελείται μονόδρομη ανάλυση διασποράς (ANOVA) διαχωρίζοντας τα μοντέλα βάσει του median των ZS επιδόσεων· δεν παρατηρείται στατιστικά σημαντική διαφορά ($F = 3.05$, $p = 0.12$).

Τέλος, μέσω γραμμικής παλινδρόμησης αξιολογείται η εξάρτηση μεταξύ ZS απόδοσης και PAKTON: η κλίση είναι μόλις 0.44, δηλαδή κάθε μονάδα βελτίωσης στο baseline μεταφράζεται σε λιγότερο από μισή μονάδα εντός

PAKTON, δείχνοντας συμπίεση διαφορών.

Συνολικά, τα ευρήματα επιβεβαιώνουν ότι το PAKTON μειώνει δραστικά τις διαφορές απόδοσης μεταξύ LLMs, ενώ διατηρεί υψηλή ακρίβεια. Αυτό είναι ιδιαίτερα σημαντικό στον νομικό τομέα, όπου η χρήση open-source μοντέλων επιτρέπει ασφαλή και ιδιωτική ανάλυση συμβολαίων.

Απόδοση της Μονάδας Επαυξημένης Δημιουργίας μέσω Ανάκτησης (ΕΔΑ)

Διαμόρφωση. Πέραν του ContractNLI, χρησιμοποιούμε το benchmark *LegalBench-RAG* [60] για την απομόνωση της μονάδας ανάκτησης—δηλαδή των Αρχειοφύλακα και Ερευνητή—χωρίς την εμπλοκή παραγωγής απαντήσεων. Το benchmark καλύπτει τέσσερις κατηγορίες: NDAs, συμφωνίες M&A, εμπορικά συμβόλαια και πολιτικές απορρήτου, επιτρέποντας μια αξιόπιστη αποτίμηση της ανάκτησης σε ετερογενή νομικά σώματα.

Το PAKTON συγκρίνεται με τέσσερις γραμμές βάσης: Naive (500 χαρακτήρες με OpenAI embeddings), RCTS (δομική κατάτμηση [42]), Naive+Cohere (με reranking), και RCTS+Cohere. Αναλυτικά στοιχεία υπάρχουν στο Παράρτημα B.

Αποτελέσματα. Ο Πίνακας 7.2 παρουσιάζει συγκριτικά την απόδοση ανάκτησης με μετρικές precision και recall για πολλαπλά k. Σε όλες τις συλλογές, το PAKTON υπερέχει σταθερά και σημαντικά έναντι όλων των baseline. Ενδεικτικά, στο ContractNLI το Recall@1 αγγίζει το 53.14% έναντι 11.32% του ισχυρότερου baseline. Αντίστοιχα υψηλά κέρδη παρατηρούνται και στα PrivacyQA, MAUD και CUAD, με διαφορές Recall@1 που ξεπερνούν τις 20 ποσοστιαίες μονάδες.

Συγκεντρωτικά, το PAKTON παρουσιάζει πάνω από πενταπλάσια αύξηση στο Recall@1 (26.77% έναντι 4.94%). Αυτά τα αποτελέσματα είναι κρίσιμης σημασίας στον νομικό τομέα, όπου η ανάκτηση μη σχετικών αποσπασμάτων μπορεί να οδηγήσει σε εσφαλμένα ή αθεμελίωτα συμπεράσματα—ιδιαίτερα σε έγγραφα με αντικρουόμενες ρήτρες ή εξαιρέσεις που απαιτούν ακριβή συμφραζόμενα.

3.3.2 Ποιοτικά Αποτελέσματα

Σε αντίθεση με τις ποσοτικές μετρικές, οι οποίες επικεντρώνονται σε μετρήσιμες παραμέτρους όπως η ακρίβεια ή η ποιότητα ανάκτησης, η ποιοτική αξιολόγηση εστιάζει στην αντιλαμβανόμενη ποιότητα και ερμηνευσιμότητα των παραγόμενων απαντήσεων. Αυτή η αξιολόγηση αντανακλά τον βαθμό στον οποίο το σύστημα υποστηρίζει τη λήψη αποφάσεων σε πραγματικά σενάρια, ιδιαίτερα στον νομικό τομέα όπου η σαφήνεια, η αιτιολόγηση και η πρακτική χρησιμότητα είναι καθοριστικής σημασίας. Ουσιαστικά, τα αποτελέσματα αυτά αποτυπώνουν πώς οι τελικοί χρήστες—ειδικά οι μη ειδικοί—ερμηνεύουν και αποτιμούν τις εξόδους του συστήματος, προσφέροντας ποιοτική εικόνα της αποτελεσματικότητας πέρα από τους αριθμητικούς δείκτες. Συγκριτικά παραδείγματα εξόδων των PAKTON και GPT-4o παρατίθενται στο Παράρτημα F.

Διαμόρφωση Πειράματος. Για την αξιολόγηση της πρακτικής χρησιμότητας του PAKTON, διεξάγεται μελέτη χρήστη με 15 ερωτήσεις-ορόσημα νομικού περιεχομένου που συντάχθηκαν από πέντε νομικούς και έναν Δικαστή του Αρείου Πάγου ώστε να αντανακλούν αυθεντικό νομικό συλλογισμό. Στην πλατφόρμα *Prolific*[1], οι συμμετέχοντες συγκρίνουν σε ζεύγη απαντήσεις των PAKTON και ChatGPT[2] βάσει εννέα κριτηρίων: ερμηνευσιμότητα, τεκμηρίωση, νομική/συμφραζόμενη κατανόηση, χειρισμός ασάφειας, αναγνώριση κενών, συντομία, συνοχή, συνάφεια και πληρότητα. Για κάθε κριτήριο, επέλεγαν την καλύτερη απάντηση και αιτιολογούσαν την επιλογή τους, με δυνατότητα επιλογής «Καμία» ή «Δεν είμαι σίγουρος» ώστε να διασφαλιστεί η αξιοπιστία. Το ChatGPT χρησιμοποιείται για την σύγκριση λόγω της ευρείας χρήσης του και, για λόγους δικαιοσύνης, αξιοποιείται RAG όταν παρέχονται εξωτερικά έγγραφα [57]. Επιπλέον λεπτομέρειες παρέχονται στο Παράρτημα D.

Αποτελέσματα. Εξήντα συμμετέχοντες ολοκλήρωσαν την αξιολόγηση, συγκρίνοντας τις απαντήσεις των PAKTON και ChatGPT ως προς 9 κριτήρια σε 15 νομικά ερωτήματα, αποδίδοντας συνολικά 540 ατομικές κρίσεις. Όπως φαίνεται στο Σχήμα 3.3.1, το PAKTON προτιμάται έναντι του ChatGPT στην πλειονότητα των διαστάσεων αξιολόγησης, μια τάση που παραμένει συνεπής σε όλα τα εξεταζόμενα σενάρια. Η μεγαλύτερη υπεροχή καταγράφεται στην «Πληρότητα», όπου το PAKTON παρέχει σταθερά πιο περιεκτικές απαντήσεις.

---

[1] https://www.prolific.com

[2] gpt-4o-2024-11-20

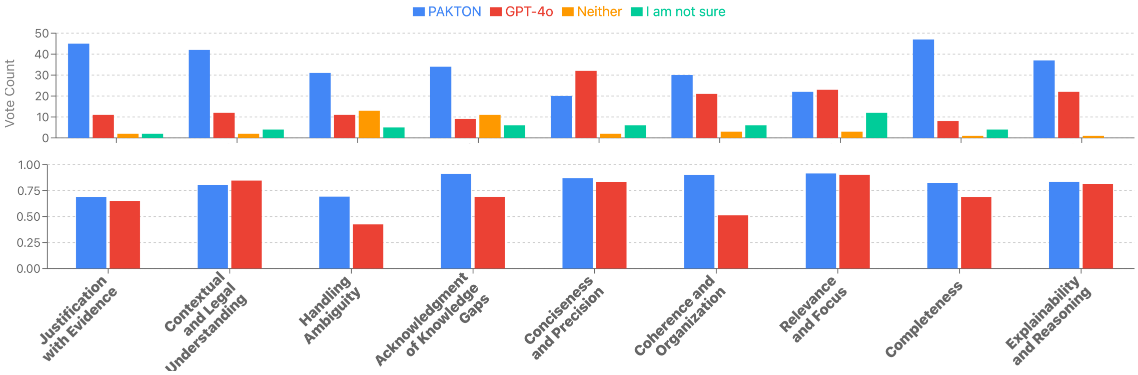


Figure 3.3.1: Συγκριτική ανάλυση PAKTON και GPT-4o. Το άνω διάγραμμα απεικονίζει τις ανθρώπινες προτιμήσεις βάσει εννέα κριτηρίων αξιολόγησης, συγκεντρωτικά για όλες τις ερωτήσεις. Το κάτω διάγραμμα παρουσιάζει τα αποτελέσματα G-EVAL για τα ίδια κριτήρια, συγκεντρωτικά για όλες τις εξόδους στο ContractNLI.

Διακρίνεται επίσης στην «Ερμηνευσιμότητα και Συλλογιστική», προσφέροντας πιο σαφείς και βηματικές αιτιολογήσεις—ιδιαίτερα σημαντικές για μη ειδικούς χρήστες.

Για την αξιολόγηση της διακριτικής ικανότητας των συμμετεχόντων, συμπεριλήφθηκαν ανταγωνιστικά κριτήρια όπως «Πληρότητα» έναντι «Συντομίας και Ακρίβειας», αναγνωρίζοντας τη φυσική ένταση μεταξύ εκτενών και συνοπτικών απαντήσεων. Όπως αναμενόταν, το PAKTON υπερέχει στην πληρότητα, ενώ το ChatGPT προτιμάται για τη συντομία. Στο κριτήριο «Συνάφεια και Εστίαση», τα δύο μοντέλα παρουσιάζουν συγκρίσιμες επιδόσεις, αντανακλώντας την ισορροπία μεταξύ εύρους και εξειδίκευσης. Τα ευρήματα αυτά επιβεβαιώνουν τον βασικό σχεδιαστικό στόχο του PAKTON για δημιουργία αναφορών τύπου έκθεσης, με το παρατηρούμενο trade-off πληρότητας-ακρίβειας να θεωρείται αναμενόμενο και δικαιολογημένο.

3.3.3 Αξιολόγηση μέσω LLMs

Η ποιότητα των απαντήσεων κάθε συστήματος αξιολογείται περαιτέρω μέσω ενός μεγάλου γλωσσικού μοντέλου (LLM), όπως αναλύεται στην παρούσα ενότητα.

Διαμόρφωση Πειράματος. Για την εκτεταμένη αξιολόγηση του PAKTON, χρησιμοποιούμε το G-EVAL [49], ένα ευρέως χρησιμοποιούμενο πλαίσιο αξιολόγησης «LLM-as-a-Judge» [51, 82, 14] για την εκτίμηση της ποιότητας παραγωγής φυσικής γλώσσας με βάση τα ίδια κριτήρια με τη μελέτη χρήστη. Αξιολογήθηκαν 102 τυχαία επιλεγμένα δείγματα από το σύνολο ContractNLI [38], συγκρίνοντας τα αποτελέσματα του PAKTON και του GPT-4o με τις ίδιες εισόδους. Το G-EVAL παρήγαγε βαθμολογίες ανά κριτήριο, επιτρέποντας λεπτομερή ανάλυση ποιότητας, ερμηνευσιμότητας και συλλογιστικής.

Αποτελέσματα. Τα αποτελέσματα για όλα τα δείγματα και κριτήρια αξιολόγησης επιβεβαιώνουν την υπεροχή του PAKTON έναντι του GPT-4o σε οκτώ από τις εννέα διαστάσεις (βλ. Σχήμα 3.3.1). Σημαντική υπεροχή παρατηρείται στην Ερμηνευσιμότητα και Συλλογιστική, στην Τεκμηρίωση μέσω Αποδείξεων, στην Πληρότητα και στον Χειρισμό Ασάφειας. Τα αποτελέσματα αυτά συνάδουν με τους στόχους σχεδιασμού του PAKTON, που δίνουν προτεραιότητα σε λεπτομερείς και αιτιολογημένες απαντήσεις με ρητή επεξεργασία νομικού περιεχομένου.

Στα κριτήρια «Συντομία και Ακρίβεια» και «Συνάφεια και Εστίαση», τα δύο μοντέλα παρουσιάζουν συγκρίσιμη απόδοση—αναμενόμενο εύρημα, καθώς αυτές οι διαστάσεις συχνά έρχονται σε αντίθεση με την Πληρότητα, έναν τομέα στον οποίο το PAKTON υπερέχει σημαντικά έναντι του GPT-4o, αναδεικνύοντας τον εγγενή συμβιβασμό μεταξύ συντομίας και βάθους.

Το μοναδικό κριτήριο στο οποίο το PAKTON υπολείπεται του GPT-4o είναι η Κατανόηση Νομικού και Συμφραζόμενου Περιεχομένου. Αν και αυτό φαίνεται εκ πρώτης όψεως παράδοξο, μια πιο προσεκτική εξέταση της συλλογιστικής του G-EVAL δείχνει ότι οι απαντήσεις του PAKTON συχνά αναγνωρίζουν ρητά την ύπαρξη κενών γνώσης σε περιπτώσεις αβεβαιότητας. Αν και αυτό θεωρείται επιθυμητό από την οπτική της διαφάνειας, το πλαίσιο αξιολόγησης το ερμηνεύει ως ένδειξη περιορισμένης κατανόησης, με αποτέλεσμα χαμηλότερες βαθμολογίες στη συγκεκριμένη διάσταση (Παράρτημα E).

3.4 Συμπεράσματα και Μελλοντικές Επεκτάσεις

Συμπεράσματα

Η παρούσα εργασία παρουσιάζει το **PAKTON**, ένα ανοιχτού κώδικα, επεκτάσιμο και πολυπρακτορικό σύστημα σχεδιασμένο ειδικά για την ανάλυση νομικών συμβολαίων. Σε αντίθεση με μαύρα κουτιά νομικής τεχνητής νοημοσύνης, το PAKTON βασίζεται σε μία διαφανή και ερμηνεύσιμη αρχιτεκτονική που απαρτίζεται από τρεις διακριτούς αλλά συνεργαζόμενους πράκτορες: τον *Archivist* για την προεπεξεργασία, τον *Researcher* για ανάκτηση γνώσης με χρήση γραφημάτων, και τον *Interrogator* για επαναληπτική πολυσταδιακή συλλογιστική. Μέσω ενός στοιχείου Επαυξημένης Δημιουργίας μέσω Ανάκτησης (ΕΔΑ), το σύστημα παράγει εμπεριστατωμένες, τεκμηριωμένες και εξηγημένες απαντήσεις σε νομικά ερωτήματα.

Το PAKTON ξεχωρίζει χάρη στον σχεδιασμό του με βάση την αρθρωτότητα, τη διαφάνεια και την προσαρμοστικότητα. Σε πειραματικά σενάρια με ποσοτικές και ποιοτικές αξιολογήσεις, πέτυχε ανώτερες επιδόσεις συγκριτικά με γενικά και ειδικά μοντέλα, σημειώνοντας κορυφαία αποτελέσματα σε σύνολα δεδομένων όπως τα ContractNLI και LegalBench-RAG. Οι ανθρώπινες αξιολογήσεις από νομικούς και αυτόματες αξιολογήσεις (π.χ. G-EVAL) επιβεβαιώνουν την ανωτερότητα του συστήματος ως προς την πληρότητα, τη θεμελίωση και την εξηγήσιμη φύση των απαντήσεων.

Ιδιαίτερη σημασία έχει το γεγονός ότι το PAKTON παρουσιάζει συνέπεια στην απόδοσή του ανεξαρτήτως του υποκείμενου ΜΓΜ. Αυτό επιτρέπει τη χρήση πιο ελαφρών, ανοιχτού κώδικα μοντέλων χωρίς ουσιαστική απώλεια ποιότητας, ευνοώντας υλοποιήσεις σε ευαίσθητα περιβάλλοντα με απαιτήσεις σε ιδιωτικότητα και διαφάνεια.

Μελλοντικές επεκτάσεις

Μελλοντικές κατευθύνσεις περιλαμβάνουν:

- **Ενοποίηση συλλογισμών από πολλαπλά ΜΓΜ**, ώστε να αξιοποιείται η ποικιλία γνώσεων και στρατηγικών συλλογισμού μέσω τεχνικών όπως weighted voting ή meta-reasoning.
- **Πολυπρακτορική επιχειρηματολογία**, με χρήση πολλών Interrogators που υιοθετούν διαφορετικές ερμηνευτικές στρατηγικές και παράγουν διαλεκτικές απαντήσεις.
- **Δυναμική ενσωμάτωση νομικής γνώσης** από επικαιροποιημένες βάσεις δεδομένων, κανονισμούς και νομολογία.
- **Πολυγλωσσική και δια-δικαιοδοτική προσαρμογή** με εργαλεία μετάφρασης, νομική αναγνώριση πλαισίου και cross-lingual embeddings.
- **Νέα νομικά ευθυγραμμισμένα πλαίσια αξιολόγησης**, μέσω σχολιασμού ειδικών, συγκριτικών προσεγγίσεων και προσομοιώσεων.

Εν κατακλείδι, το PAKTON δεν αποτελεί απλώς ένα εργαλείο για ανασκόπηση συμβολαίων, αλλά μία ευέλικτη, επεκτάσιμη πλατφόρμα νομικής τεχνητής νοημοσύνης με προοπτική υψηλής αξιοπιστίας και δικαιοσύνης.

3.5 Περιορισμοί και Ηθική

Παρότι το PAKTON παρουσιάζει εξαιρετικές επιδόσεις, συνοδεύεται από σημαντικούς περιορισμούς και ηθικές προκλήσεις που απαιτούν προσεκτική αξιολόγηση.

Τεχνικοί Περιορισμοί

- **Γλωσσική εμβέλεια**: Το σύστημα έχει δοκιμαστεί μόνο σε αγγλικά συμβόλαια. Η πολυγλωσσική γενίκευση απαιτεί προσαρμογή.

- **Κάλυψη συμβολαίων και δικαιοδοσιών**: Δεν έχει αξιολογηθεί σε όλο το εύρος νομικών εγγράφων ή νομικών πλαισίων.
- **Καθυστέρηση και κόστος**: Η πολυσταδιακή λογική αυξάνει τον χρόνο και το υπολογιστικό κόστος.
- **Διαφάνεια vs αποτελεσματικότητα**: Εξηγήσιμες απαντήσεις μπορεί να είναι λιγότερο σύντομες ή ελαφρώς λιγότερο ακριβείς.
- **Δομική γενίκευση parsing**: Όταν η μορφή του εγγράφου αποκλίνει από το πρότυπο, μειώνεται η ακρίβεια της ανάλυσης.

Ηθικά Ζητήματα

- **Υπερβολική εμπιστοσύνη**: Μη ειδικοί χρήστες μπορεί να βασιστούν υπέρμετρα στις απαντήσεις. Το σύστημα πρέπει να προβάλλεται ως εργαλείο υποστήριξης, όχι υποκατάστατο νομικής συμβουλής.
- **Προσβασιμότητα**: Ενώ προωθείται η ανοικτή διάθεση, η τεχνική υποδομή παραμένει εμπόδιο για ευάλωτες ομάδες.
- **Ασφάλεια και κακή χρήση**: Υπάρχει κίνδυνος επιθέσεων ή καταχρήσεων (π.χ. prompt injection). Απαιτείται συνεχής επιτήρηση.
- **Ηθική αξιολόγηση**: Οι αξιολογήσεις έγιναν με πλήρη ενημέρωση, χωρίς συλλογή προσωπικών δεδομένων, διασφαλίζοντας δεοντολογία και δικαιοσύνη.

Η αντιμετώπιση αυτών των ζητημάτων αποτελεί προϋπόθεση για την υπεύθυνη και ασφαλή υλοποίηση του PAKTON σε πραγματικές νομικές εφαρμογές.

Chapter 4

Introduction

Contracts are among the most foundational legal documents, governing a wide range of personal, commercial, and governmental relationships. Yet, analyzing contracts remains a complex and time-consuming task that typically requires legal expertise. As a result, contract review is often inaccessible to the general public and remains demanding even for professionals. Research by World Commerce & Contracting (WorldCC) reveals that organizations lose an average of 9.2% of their annual revenue due to contract mismanagement, with that figure rising to 15% for larger enterprises [89]. In parallel, the Institute for Supply Management (ISM) has reported that a typical Fortune 1000 company manages between 20,000 and 40,000 active contracts at any given time[35], while even simple agreements can take over a week to approve. These challenges highlight the growing need for better support in understanding contracts and reducing the time required for manual review.

Recent advancements in large language models (LLMs) show promise in natural language understanding[29] [58], question answering, and document summarization [1, 6]. While LLMs are powerful, their application in specialized domains such as legal contract analysis presents unique challenges [26, 24, 36], including justifying decisions with clear reasoning and referencing both the contract and relevant external sources [92]. Contract documents also exhibit several peculiarities that require specialized handling. They often contain complex legal terminology, calling for domain-specific language understanding. Overlapping or even contradictory clauses appear frequently [52, 34, 2], requiring robust clause retrieval and conflict resolution [2, 95]. Similarly, exceptions and references to different document parts also request robust retrieval mechanisms. Ambiguous phrasing and multiple interpretations are common, making careful contextual analysis essential. Additionally, legal differences across jurisdictions necessitate consultation with external legal databases to maintain precision and relevance.

Retrieval-Augmented Generation (RAG)[28, 25, 79, 27] directly addresses these limitations by integrating targeted retrieval of internal and external documents into the LLM generation process, grounding outputs in domain-specific, verifiable evidence. This approach enhances not only factual accuracy but also transparency and explainability—critical attributes for legal contract analysis. Moreover, RAG plays a pivotal role in the legal domain, where the inability to retrieve relevant spans can compromise reasoning and produce unsupported conclusions[60]. Furthermore, given that legal contracts frequently contain sensitive or confidential information, proprietary models are often unsuitable, necessitating open-source alternatives capable of operating under limited computational resources. Consequently, the development of efficient mechanisms for contract analysis becomes imperative.

Within this context, we introduce PAKTON[1] [63], a multi-agent framework designed to analyze contract documents and provide explainable, legally grounded answers to user queries along with a comprehensive report. Inspired by [67] the proposed system is composed of three collaborative agents: (1) the *Archivist*, which interacts with the user and manages structured document input; (2) the *Researcher*, which retrieves relevant internal and external information using hybrid and graph-aware retrieval techniques; and (3) the *Interrogator*, which engages in multi-step reasoning to iteratively refine the final report.

---

[1] *PAKTON* comes from the ancient Greek word that means agreement or contract, related to the Latin "pactum".

PAKTON departs from black-box models by prioritizing transparency, progressive refinement, and grounded justifications. It generates structured legal reports with topic summaries, legal reasoning, key findings, and precise citations to contract clauses and external sources, while explicitly flagging knowledge gaps to avoid unsupported claims. All components are modular, open source, lightweight, and support on-premise deployment. We benchmark PAKTON on five contract analysis tasks using ten criteria—combining automated and human evaluation—and find that it substantially surpasses general-purpose LLMs in retrieval accuracy and explainability.

The outline of this thesis is as follows:

- We begin by presenting the necessary background in foundational Natural Language Processing concepts, including Large Language Models, Prompting Techniques, Retrieval-Augmented Generation and multi-agent systems.
- We then offer a comprehensive review of related work within the domain, highlighting key distinctions and contributions that differentiate our approach.
- Finally, we describe the architecture of the proposed PAKTON framework in detail and evaluate its performance against established baselines across multiple evaluation metrics and datasets.

Chapter 5

Background

Before embarking on the experimental component of this thesis, it is crucial to first establish a robust conceptual and methodological foundation. This chapter provides a comprehensive overview of key advancements in Artificial Intelligence (AI), with particular emphasis on Large Language Models (LLMs) and Transformer-based architectures, technologies that form the cornerstone of the research presented herein. A clear understanding of these underlying principles is vital for grasping the design, execution, and interpretation of the experiments that follow.

Transformer-based Architectures and Large Language Models Transformer-based architectures, particularly those that power LLMs, have revolutionized the field of Natural Language Processing (NLP) by enabling the modeling of intricate dependencies within language data. These models exhibit remarkable proficiency in tasks such as text generation, semantic inference, and logical reasoning. Gaining a clear understanding of how Transformers operate and how their architecture supports reasoning is a prerequisite for assessing their performance in the experimental tasks presented later.

Training Methodologies for Large Language Models A fundamental aspect of this research involves the *training* of LLMs. Training refers to the iterative adjustment of a model's parameters to optimize its performance on targeted tasks. Although most LLMs are pre-trained on extensive corpora, they often require additional task-specific fine-tuning to adapt to specialized domains or problem types. This section will survey standard training practices, such as supervised fine-tuning, as well as advanced strategies including reinforcement learning from human feedback (RLHF) and continual learning.

Prompt Engineering and Task Conditioning Beyond conventional training paradigms, *prompting* has emerged as a powerful mechanism for controlling and conditioning the behaviour of LLMs. Prompt engineering involves the design of input sequences that elicit desired responses from a model without necessitating additional training. This technique has gained prominence due to its efficiency and flexibility, especially in scenarios where retraining is computationally prohibitive. In this thesis, we analyze a variety of prompting methods—ranging from zero-shot and few-shot prompting to more structured techniques such as chain-of-thought (CoT) prompting—and compare their performance against the performance of our framework.

Retrieval-Augmented Generation (RAG) Retrieval-Augmented Generation (RAG) is a hybrid architecture that enhances language models by integrating them with an external information retrieval system. In this framework, the model dynamically retrieves relevant documents or context passages from a corpus and incorporates them into its generative process. This approach mitigates the limitations of static model knowledge and enables more informed, context-sensitive responses. RAG is particularly advantageous in tasks requiring factual precision or domain-specific knowledge, and it plays a key role in experiments involving question answering and reasoning under uncertainty in this thesis.

Agents and Multi-Agent Systems In recent AI research, there has been growing interest in modeling problem-solving as a distributed process among multiple interacting agents. *Agent-based systems* encapsulate autonomous units—each with specific goals, capabilities, and policies—that collaborate or compete to achieve complex objectives. Extending this concept, *multi-agent systems* (MAS) introduce the possibility of decentralized reasoning, parallel exploration, and role specialization. Within this thesis, we explore how agent-based frameworks, when integrated with LLMs, can emulate collaborative problem-solving and enhance performance on reasoning tasks that benefit from diverse perspectives or decomposed cognitive strategies.

Conclusion This chapter has presented a comprehensive overview of the key concepts underpinning this research, including Transformer-based architectures, training methodologies, prompting strategies, Retrieval-Augmented Generation, and agent-based reasoning systems. By situating our experimental work within these theoretical frameworks, we aim to equip the reader with the requisite background to critically evaluate the approaches and results discussed in subsequent chapters.

5.1 Large Language Models (LLMs)

5.1.1 Background

Language Models (LMs) are computational models that have the capability to understand and generate human language. More specifically, a language model is a probability distribution over word sequences, having the ability to predict the likelihood of these sequences or generate new text based on a given input.

The fundamental methodology for probabilistic language modeling since 1980's has been **n-grams models**. These models are grounded on the Markov chain rule, assuming that the probability of the coming word in a word sequence depends only on a fixed size window of previous words. Hence a bigram model considers one previous word, a trigram two, and in general a n-gram n-1 previous words.

For example a bigram language model models the probability of the sequence $w_1, w_2, \dots w_n$ as:

$$P(w_1, w_2, \dots w_n) = P(w_2, w_1) \cdot P(w_3|w_2) \cdot \dots \cdot P(w_n|w_{n-1}) \tag{5.1.1}$$

where the conditional probability $P(w_k|w_{k-1})$ can be estimated the proportion of occurrences of the word w_{k-1} followed by the word w_k in the corpora. The estimation of these probabilities constitutes the training of an n-gram model on text corpora in one or more languages. However, given that a language can be used to express an infinite variety of valid sentences, an n-gram model struggles to assign non-zero probabilities to word sequences that may never be encountered in the training corpora. To address this problem, a variety of smoothing techniques has been proposed over the years.

Subsequently, the advancements in the development of neural networks suggest the neural language models, which was first introduced with simple **feedforward neural language models** by [9]. A feedforward neural language model is a simple feedforward network that takes a sequence of previous words as input and gives a probability distribution over possible next words as output. Therefore, similarly to a n-gram LM, the feedforward neural LM is trained to predict the probability of a word considering the n-1 previous words.

A major difference between n-grams and feedforward neural language models is found in the way the two models represent the sequence of input words. An n-gram model assigns the word identity i to each word w_i included in the prior word sequence which should be represented. On the contrary, a feedforward neural language model mainly uses an embedding vector for each word. Thus, the latter generalizes better to unseen word sequences of the test set.

While feedforward language models introduced many of the foundational concepts of neural language modeling, modern neural language models have advanced significantly by utilizing more powerful architectures. In this thesis, we will focus on analyzing the state-of-the-art (SotA) architecture: Transformer networks, which have revolutionized the field with their superior ability to model complex dependencies in language.

5.1.2 Transformer

Large Language Models (LLMs) are advanced language models with vast parameter sizes and remarkable learning capabilities that are typically pretrained on large unstructured text corpora. All modern LLMs are now built on Transformer architecture [75], which eschews recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output. The *encoder* and *decoder* are the two major components of the Transformer architecture. These components are employed in sequence-to-sequence operations like machine translation, where the encoder processes the input sequence and the decoder generates the output sequence.

The Transformer architecture is summarised below:

1. **Input Representation**: The input sequence is first embedded into continuous vector representations. Positional embeddings are added to these embeddings to provide information about the position of each token in the sequence.
2. **Encoder**: The encoder consists of a stack of identical layers. Each layer has two sub-layers:

  - *Multi-Head Self-Attention*: This sub-layer computes a weighted sum of embeddings, allowing each word to focus on different parts of the input sequence. Multiple attention heads run in parallel, capturing different relationships between words.
  - *Position-wise Feed-Forward Neural Network*: After the attention mechanism, each token's representation is passed through a position-wise feed-forward neural network. This introduces non-linearity and further refines the token representations.

   Residual connections[32], followed by layer-normalization[8] are employed around each of the sub-layer.

3. **Decoder**: The decoder also consists of a stack of identical layers, each containing three sub-layers:
   - *Masked Multi-Head Self-Attention*: This sub-layer acts similar to the corresponding encoder's sub-layer, but with a mask applied to prevent attending to future positions during training.
   - *Multi-Head Encoder-Decoder Attention*: This sub-layer focuses on the encoded input sequence, allowing the decoder to consider the relevant parts of the input during sequence generation.
   - *Position-wise Feed-Forward Neural Network*: Similar to the encoder, this sub-layer follows the attention mechanisms.

   As with the encoder, residual connections are used around each sub-layer, followed by layer-normalization.

4. **Output Generation**: The output of the final decoder layer is transformed into probability distributions over the output vocabulary using a linear transformation followed by a softmax activation. Throughout the training process, the model is fed with a word sequence as input to predict the subsequent word.

Transformer variations

Several architectural variations of the Transformer have been proposed since it was first introduced by [75]. The masking pattern used on the inputs, which acts as contextual information for the model to generate a prediction, is a key distinction between these systems.

Encoder-Decoder. As previously indicated and originally proposed, the Transformer consisted of two stacks (Fig. 5.1.1): the encoder and the decoder. The encoder processes the input sequence and generates context-rich representations, which are used by the decoder to generate the output sequence step by step. Notable pretrained language models using an encoder-decoder architecture include BART[43] and T5 [62].

Decoder-Only. While the encoder-decoder design serves as the foundational variation of the Transformer model, contemporary LLMs predominately employ a decoder-only architecture. These models have the capability to train as a conventional language model, wherein they learn to predict the next token in a given sequence. Decoder-only models lack the ability to process or represent the input sequence and output sequence separately. All tokens are treated equally during processing, and conditioning is only dependent on prior tokens due to the casual masking pattern, implying that the representation of any conditioning text is intrinsically weaker. However, this produces a simpler architecture that is well-suited to a standard autoregressive next-step-prediction pretraining objective. Notably, this architecture is the foundation of the GPT series of models [61, 12] as well as numerous other recent LLMs.

Encoder-Only. As an aside, there is an additional prevalent architectural variant that employs only a Transformer encoder layer stack. This model architecture serves as the foundation for the ubiquitous BERT [20] and its derivatives.

Transformers have revolutionized NLP by eliminating the need for convolutions or recurrence mechanisms while achieving state-of-the-art performance on various tasks. Their ability to model long-range dependencies and efficiently process sequential data using the attention mechanism has set a new standard for language models.

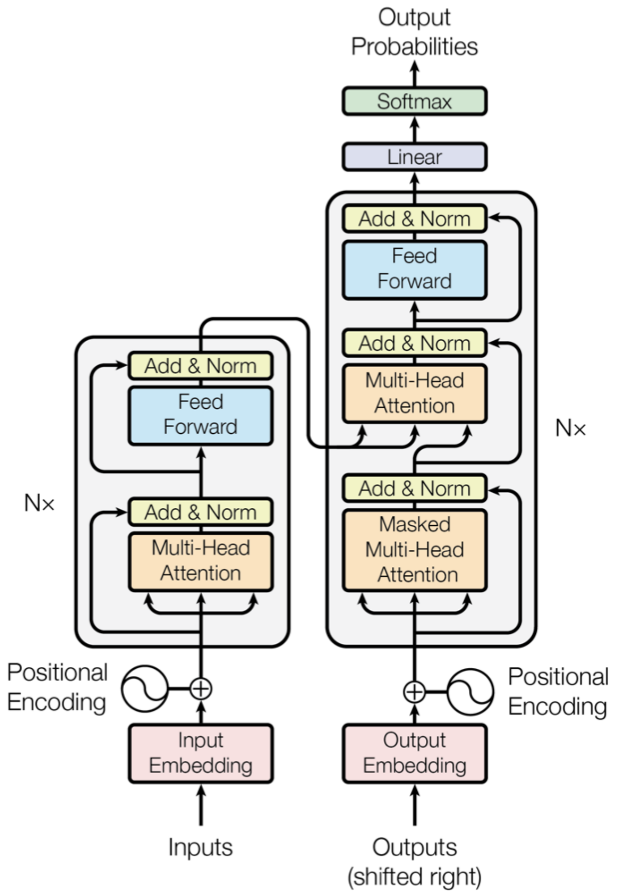


Figure 5.1.1: **The Transformer - model architecture**. The original Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder, shown in the left and right halves of figure respectively[75]

.

5.1.3 Deeper Exploration of LLM Components

The LLMs used in this work are composed of multiple components that allow them to process, generate, and understand natural language. The following sections will delve deeper into the mathematical formulations of these key components. We will analyze each of these, including tokenization, embedding layers, self-attention mechanisms, multi-head attention, position encoding, feed-forward layers, and the final output layer.

Tokenization and Embedding Layer

Tokenization: Text is split into tokens, which can be words, subwords, or characters. Let $x = (x_1, x_2, \ldots, x_n)$ represent an input sequence of n tokens.

Embedding: Each token x_i is transformed into a dense vector $e_i \in \mathbb{R}^d$ using an embedding matrix $E \in \mathbb{R}^{V \times d}$, where V is the vocabulary size and d is the embedding dimension. The embedding for the entire input sequence can be represented as:

$$E(x) = (e_1, e_2, \ldots, e_n)$$

where $e_i = E[x_i]$.

Self-Attention Mechanism

The self-attention mechanism is central to transformer-based LLMs. For each token in the input sequence, the attention mechanism computes a weighted sum of the other tokens, determining how much focus the model should place on each token when processing the sequence.

For a given input $x = (x_1, x_2, \ldots, x_n)$, the attention mechanism relies on three matrices: the query (Q), key (K), and value (V) matrices. These matrices are computed as:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V$$

where $X \in \mathbb{R}^{n\times d}$ is the input embedding matrix, and $W^Q, \in \mathbb{R}^{d\times d_q}$, $W^K, \in \mathbb{R}^{d\times d_k}$, $W^V, \in \mathbb{R}^{d\times d_v}$ are learned parameter matrices that project the input embeddings into the query, key, and value spaces, respectively.

The attention scores are computed by taking the dot product of the query with the keys, scaled by the square root of the dimensionality d_k, followed by a softmax operation:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

This equation computes a weighted sum of the value vectors, where the weights are determined by the similarity between the query and key vectors. The softmax function ensures that the attention weights sum to 1, giving a probabilistic interpretation to the weights. An visual representation of the self-attention mechanism can be seen in Figure 5.1.2.

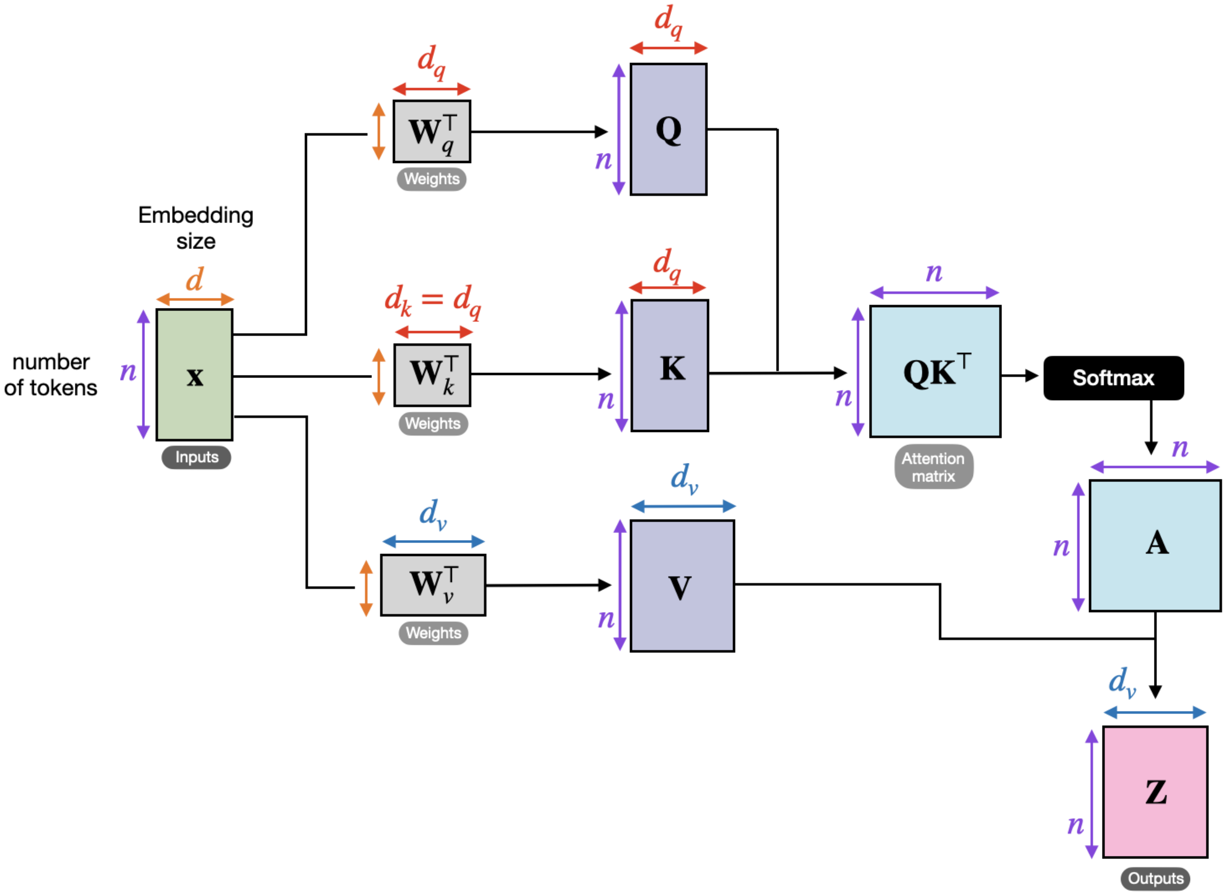


Figure 5.1.2: Self-Attention Mechanism

Multi-Head Attention

Instead of computing a single set of attention scores, LLMs use multi-head attention to capture different types of relationships between tokens. In multi-head attention, multiple attention mechanisms (heads) are run in parallel, and their outputs are concatenated:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^O$$

where each head is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

and $W^O \in \mathbb{R}^{hd_k \times d}$ is a learned output projection matrix. The use of multiple heads allows the model to attend to different parts of the input in different ways, increasing its representational power.

Position Encoding

Transformers do not have an inherent notion of word order, so position encoding is added to capture the sequential nature of the input. A positional encoding matrix $P \in \mathbb{R}^{n \times d}$ is added to the input embeddings. The positional encoding is typically defined using sinusoidal functions:

$$P_{i,2j} = \sin\left(\frac{i}{10000^{2j/d}}\right), \quad P_{i,2j+1} = \cos\left(\frac{i}{10000^{2j/d}}\right)$$

where i is the position and j is the dimension. This encoding allows the model to differentiate between positions in the sequence while maintaining generalization to longer or shorter sequences.

Feed-Forward Layer

Each transformer block includes a feed-forward network (FFN) applied independently to each position. The FFN consists of two linear transformations with a non-linearity in between:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

where $W_1 \in \mathbb{R}^{d \times d_{ff}}$ and $W_2 \in \mathbb{R}^{d_{ff} \times d}$ are learned weight matrices, $b_1 \in \mathbb{R}^{d_{ff}}$ and $b_2 \in \mathbb{R}^d$ are bias terms, and $\max(0, \cdot)$ represents the ReLU activation function. The FFN is applied separately to each token in the sequence.

Final Output Layer

The final output of the transformer model is passed through a linear transformation and softmax layer to generate probabilities for each class or token prediction. Let h_i represent the hidden state of the i-th token after the transformer layers. The probability distribution over the vocabulary is computed as:

$$P(x_i|h_i) = \text{softmax}(h_i W^T + b)$$

where $W \in \mathbb{R}^{V \times d}$ is the learned weight matrix, and $b \in \mathbb{R}^V$ is the bias term.

5.2 Training Techniques

Machine Learning (ML) techniques can be broadly categorized based on the type of data used during training and the level of supervision provided. These categories include supervised, unsupervised, and reinforcement learning, with hybrid approaches such as semi-supervised and self-supervised learning blending aspects of both supervised and unsupervised methods. To provide a comprehensive context for the techniques used in this thesis, it is essential to first explore these learning paradigms before diving deeper into the specific approach of supervised fine-tuning, which will be analyzed in detail in the following sections.

5.2.1 Overview

Supervised Learning Supervised learning is one of the most common ML techniques, where models are trained on labeled data. Each input feature vector x is paired with a known target y, and the model learns to map inputs to outputs by estimating the conditional probability distribution $p(y|x)$. Supervised learning is used for tasks such as classification, regression, and forecasting, where the relationships between inputs and outputs are explicitly defined. While highly effective in many domains, supervised learning requires large amounts of labeled data, which can be resource-intensive to obtain, particularly for complex tasks.

Unsupervised Learning In unsupervised learning, models are trained on data without labeled outputs. The goal is to learn the underlying structure of the data by modeling the distribution $p(x)$. This approach is commonly used for tasks like clustering, dimensionality reduction, and anomaly detection, where the model uncovers patterns and relationships within the data without explicit supervision. While unsupervised learning can provide valuable insights, it lacks the direct feedback present in supervised learning, making it harder to evaluate and interpret the results.

Semi-supervised Learning Semi-supervised learning bridges the gap between supervised and unsupervised learning by utilizing a small set of labeled data alongside a larger set of unlabeled data. This is particularly useful in cases where labeled data is scarce or expensive to collect, but unlabeled data is abundant. By leveraging the unlabeled data to help the model generalize better, semi-supervised learning aims to improve performance over a purely supervised approach. Applications of semi-supervised learning include tasks like link prediction in graphs and fraud detection, where having even a limited amount of labeled data can significantly enhance model performance [87].

Self-supervised Learning Self-supervised learning is an increasingly popular approach that lies between supervised and unsupervised learning. In this paradigm, the model is trained on unlabeled data by solving pretext tasks designed to learn useful representations. For example, models can be tasked with predicting missing parts of a sequence or reconstructing corrupted data. The learned representations can then be transferred to downstream tasks with minimal fine-tuning. Self-supervised learning has proven particularly effective in fields such as natural language processing and computer vision, where massive amounts of unlabeled data are available [50]. This technique forms the foundation of many large language models, which are pretrained using self-supervised methods before being fine-tuned for specific tasks.

Reinforcement Learning Reinforcement learning involves training an agent to interact with an environment and make decisions that maximize a reward signal. Unlike the static datasets used in supervised and unsupervised learning, the agent in reinforcement learning continuously learns from the outcomes of its actions. This method has been successfully applied in areas such as robotics, game playing, and decision-making systems. However, reinforcement learning is beyond the scope of this thesis and will not be explored further.

5.2.2 Pre-Training and Fine-Tuning

In machine learning, the training process typically follows a two-phase approach: pre-training and fine-tuning. These stages enable models to first develop a broad understanding of data and subsequently specialize in solving specific tasks. This approach has been particularly successful in natural language processing (NLP) and other domains where data variability and task specificity are critical for achieving high performance.

Pre-Training

Pre-training is the initial phase, where a model is exposed to large, unstructured datasets, often without explicit task labels. During this stage, the model learns general patterns, features, and representations from the data, which form a solid foundation for subsequent fine-tuning. For language models, pre-training often involves tasks such as predicting masked words in a sentence (masked language modeling) or predicting the next word in a sequence (autoregressive modeling). This phase equips the model with broad domain knowledge that can be transferred to more specific tasks during fine-tuning.

The key benefit of pre-training lies in its ability to help the model build generalizable representations that can be leveraged for various downstream tasks. Pre-trained models have demonstrated superior performance across a wide range of applications due to their capacity to capture essential patterns, such as semantic relationships and syntactic structures. This enables them to adapt quickly and efficiently to specific tasks, even when task-specific data is limited.

Pre-Training Benefits in this Context

In the context of this investigation, pre-training offers several key advantages:

- **Generalization**: Pre-training on both large general datasets and task-specific datasets enhances the model's ability to generalize to unseen examples, which is particularly important given the limited size of the fine-tuning dataset.
- **Efficiency**: By leveraging pre-trained knowledge, the model requires less task-specific data to perform effectively. Instead of learning from scratch, it uses the information acquired during pre-training to adapt efficiently during fine-tuning.
- **Faster Adaptation**: Pre-training on related tasks reduces the need for extensive training iterations during fine-tuning. The model can quickly converge to optimal performance, as it is already aligned with the task requirements.

Fine-Tuning

Fine-tuning builds on pre-training by adapting the pre-trained model to a specific task. In this phase, the model is trained on a smaller, task-specific dataset, typically with labeled examples, enabling it to specialize in solving the given problem. Fine-tuning is highly effective because it allows models to transfer the general knowledge acquired during pre-training to the specific nuances of a new task. By adjusting the model's parameters to fit task-specific data, fine-tuning enables the model to better capture patterns and relationships unique to the task.

Fine-tuning is particularly valuable in cases where large amounts of task-specific data are difficult to obtain. By leveraging the broad understanding gained during pre-training, fine-tuning enhances the model's performance on small datasets, reducing the risk of overfitting and improving generalization.

Fine-Tuning with Limited Data

A common challenge in machine learning, especially for specialized tasks, is the scarcity of labeled data. Fine-tuning on small datasets can be problematic when training a model from scratch, as the model is likely to overfit, failing to generalize beyond the training set. However, fine-tuning a pre-trained model helps mitigate this issue by providing a strong initialization based on prior knowledge.

Fine-tuning pre-trained models on small datasets offers several benefits:

- **Data Efficiency**: Since the model has already learned general patterns during pre-training, fine-tuning on limited data mainly involves adapting these patterns to the task-specific nuances. This reduces the need for extensive labeled data and helps prevent overfitting.
- **Improved Generalization**: The pre-trained model's existing knowledge allows it to generalize better even with smaller datasets, as it is less dependent on the fine-tuning data alone. This leads to better performance on unseen examples compared to models trained from scratch.

Thus, in scenarios with limited data, fine-tuning pre-trained models proves advantageous by providing robust generalization and minimizing overfitting risks.

Fine-Tuning with Task-Specific Pre-Training

Beyond general pre-training, models can benefit significantly from **domain-adaptive pre-training**, where pre-training data is carefully selected to closely align with the target task. By pre-training the model on datasets similar to the downstream task, the model becomes more adept at capturing domain-specific patterns and representations.

For instance, in NLP tasks, a model pre-trained on a general corpus (e.g., Wikipedia or Common Crawl) may perform reasonably well across a variety of tasks. However, fine-tuning this model on a specialized task (e.g., commonsense reasoning or disambiguation) can be further enhanced by an intermediate pre-training step using a dataset closely related to the final task before fine-tuning.

This approach offers several advantages:

- **Transfer of Domain-Specific Knowledge**: By pre-training on data that shares domain-specific characteristics with the target task, the model captures nuances and representations closely aligned with the downstream task. This leads to faster convergence and typically results in better performance during fine-tuning.
- **Enhanced Performance**: Models that have been pre-trained on similar tasks generally outperform models that only undergo general pre-training, as they require less adaptation and can leverage the domain-specific knowledge already acquired.
- **Efficient Adaptation**: Domain-adaptive pre-training reduces the amount of adaptation required during fine-tuning. Since the model has already encountered related problems, the fine-tuning phase becomes more efficient, improving both generalization and performance.

5.3 Prompting

Prompt-based learning represents a paradigm shift in machine learning, departing from conventional supervised learning approaches. Unlike traditional methods that rely on training models to predict outputs based on inputs, prompt-based learning harnesses the power of language models to directly model the probability of text. At its core, this approach involves transforming an original input into a textual prompt by using a template wherein certain slots remain unfilled. These slots serve as placeholders for the language model to generate the missing information, resulting in a final string that informs the desired output [48].

One of the defining features of prompt-based learning is its ability to leverage pre-trained language models that have been exposed to massive amounts of raw text data. This pre-training equips the models with a robust understanding of language and enables them to infer missing information in prompts effectively. Moreover, by crafting suitable prompting functions, these models can adapt to new tasks with remarkable flexibility, often requiring only a few examples (*few-shot learning*) or even none at all (*zero-shot learning*).

This framework offers several advantages over traditional approaches. Firstly, it capitalizes on the vast amount of unlabeled text data available, enabling models to capture intricate linguistic patterns and nuances. Secondly, it empowers models to swiftly adapt to novel scenarios, making it well-suited for tasks where labeled data is scarce or costly to obtain. Overall, prompt-based learning emerges as a potent methodology, revolutionizing the landscape of machine learning with its versatility, efficiency, and scalability.

Compared to conventional fine-tuning, which expensively updates the massive LM parameters for each downstream task, prompting concatenates the inputs with an additional piece of text that guides the LM to generate the desired outputs [19].

In general, prompting has rapidly evolved as a standard technique to elicit the "hidden" capabilities of LLMs, successfully serving a vast variety of tasks in NLP and beyond [40, 39, 7, 59, 74].

5.3.1 Prompting Methodology

The primary challenge with traditional **supervised learning** is the need for extensive annotated data to train a model that estimates $P(y|x;\theta)$, where y is the label and x is the input. However, for many tasks, such annotated data can be scarce or costly to obtain. Prompt-based learning techniques in **natural language processing (NLP)** aim to address this challenge by shifting the focus towards training a language model (LM) that estimates the probability of the text $P(x;\theta)$ itself. By doing so, it enables the prediction of y without requiring large labeled datasets. Prompt-based approaches typically involve three core steps to predict the highest scoring answer, as outlined below [48].

1. Prompt Addition

In the first step, a **prompting function** is used to convert the input text into a prompt. This is done by designing a **template** that incorporates two key slots:

- **Input slot [X]**: This slot holds the input text.
- **Answer slot [Z]**: This slot is designed to hold an intermediate answer that will later be mapped to the final output y.

Once the template is created, the input slot **[X]** is filled with the input text. The goal is to structure the task in such a way that the language model can generate a relevant intermediate output by processing the prompt.

2. Answer Search

In this step, the language model searches for the answer that generates the highest probability. To achieve this:

- A **set of possible answers** is defined, typically a list of potential intermediate answers z.
- A function is then used to iteratively fill the answer slot **[Z]** in the prompt with each of the potential answers z.

After each prompt is completed, the language model measures the **probability** of the filled prompt (i.e., the probability of the text generated by the LM) and identifies the **highest-scoring answer**.

3. Answer Mapping

In some tasks, it may be necessary to map the intermediate answer z to the final output y. For example, in **sentiment analysis**, words like "excellent," "good," or "beautiful" might need to be mapped to a final output such as "++" (very positive sentiment). This ensures that the model produces the desired output for the task. However, in tasks such as text generation, this mapping step might be unnecessary, as the output directly corresponds to the generated text.

5.3.2 Prompt Categories

The prompting function can take two forms, either being the same for each input—thus called **static**—or generating a different template for each input—thus called **dynamic**. The categorization can be seen in 5.3.1. In addition to this distinction, prompts are classified into different types [48]:

- **Discrete Prompts (a.k.a. hard prompts)**: These are specific, hand-crafted text-based prompts written in human-interpretable natural language. For example, a discrete prompt for a sentiment analysis task might be: “The sentiment of the sentence [X] is [Z]," where the model fills in [Z] with appropriate sentiment labels like "positive" or "negative."
- **Continuous Prompts (a.k.a. soft prompts)**: These prompts operate directly in the embedding space of the model, rather than in human-readable text. Continuous prompts involve learnable embedding vectors that can be optimized through gradient descent.

While **soft prompts** are popular due to their compatibility with gradient descent, they come with several drawbacks:

- **Lack of interpretability**: Embedding vectors are difficult for humans to comprehend, making soft prompts less interpretable.
- **Incompatibility with other LLMs**: Soft prompts are often incompatible with other large language models and can’t be easily transferred, as embedding spaces may differ across models [85].
- **Costly to use**: Soft prompts are typically unavailable for models accessed only through inference APIs, and their use requires access to the model’s internal embedding space, making them generally more expensive or impractical to use in deployed environments.

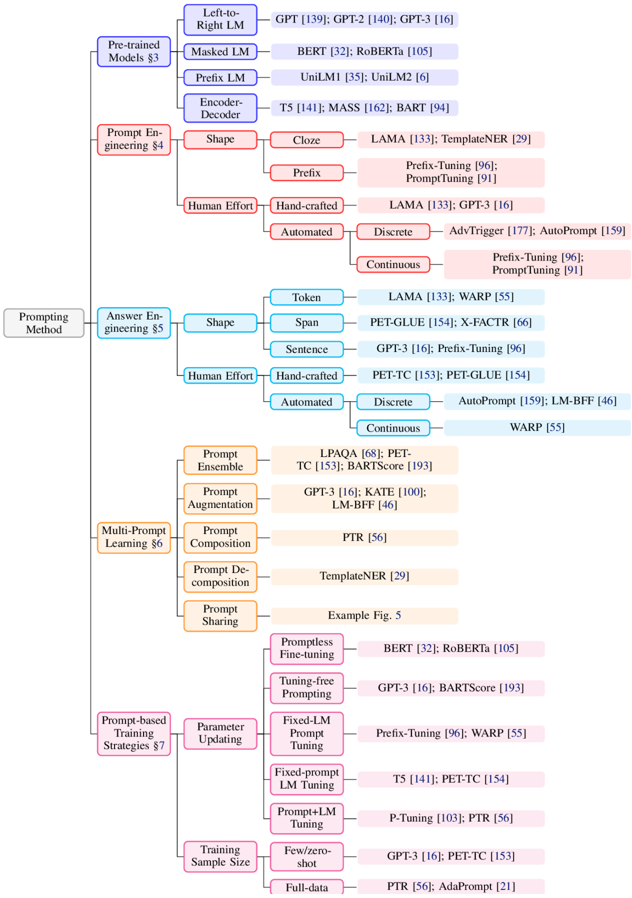


Figure 5.3.1: Prompting Typology [48]

5.3.3 Prompting Techniques

Instruction tuning, where models can learn new tasks based solely on natural language instructions, has been widely proposed and used in various applications [23]. Some of the most prominent methods include zero-shot learning, one-shot learning, few-shot learning, and chain-of-thought prompting.

In **zero-shot prompting**, the model is given an instruction directly without any examples. The model uses its pre-trained knowledge to complete the task based purely on the instruction.

One-shot learning and **few-shot learning** extend this idea by providing one or several examples to the model. In one-shot learning, the model is presented with a single example of the task. In few-shot learning, the model is given multiple examples (typically k examples) of input-output pairs (context and completion), followed by a final input (context) for which the model must generate the appropriate output (completion).

Few-shot examples can be selected using various criteria:

- **Random selection**: Examples can be chosen randomly from the training data.
- **Semantic similarity**: Examples that are semantically similar to the new task or context can be selected to guide the model more effectively.
- **Diversity**: Choosing a diverse range of examples can help the model generalize better to different inputs.

Chain-of-thought prompting is a method in which the model is guided through a series of intermediate reasoning steps. This approach has been shown to significantly improve the ability of large language models (LLMs) to perform complex reasoning tasks, as it encourages the model to break down problems into smaller steps [84].

Few-shot Example Selection Based on Semantic Similarity

When selecting few-shot examples, one of the most effective approaches is to choose examples based on **semantic similarity** between the context of the new task and the existing examples. This selection is typically done using **embeddings** and **cosine similarity** to measure the closeness between different pieces of text.

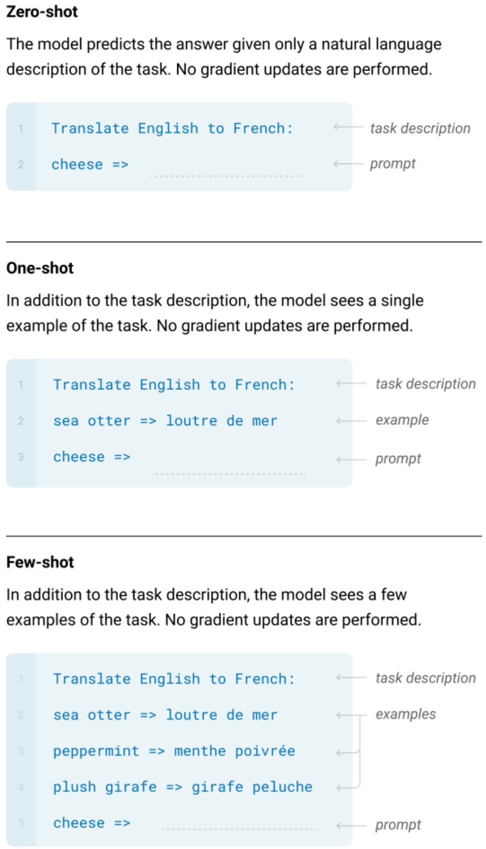
Embeddings are dense vector representations of words, sentences, or even entire documents. Unlike traditional sparse representations (such as one-hot encoding), embeddings capture the semantic relationships between different linguistic units by mapping them into a continuous vector space. In this space, words or phrases that are semantically similar are positioned closer to each other.

Pre-trained models, like **BERT** or **GPT**, generate embeddings that capture a rich set of features for each word or sentence, derived from vast amounts of training data. These embeddings can be used to compare different text inputs and identify the most relevant examples based on their semantic similarity.

To select few-shot examples based on **semantic similarity**, the process typically involves the following steps:

1. **Embedding Extraction**: First, embeddings are generated for the new task's context (input) and for the potential examples from the dataset.
2. **Cosine Similarity Calculation**: The **cosine similarity** between the embeddings of the new task context and the example contexts is computed. Cosine similarity measures the cosine of the angle between two vectors in a multi-dimensional space. It ranges from -1 (completely dissimilar) to 1 (completely similar). The formula for cosine similarity between two vectors A and B is:

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{||A|| \, ||B||}$$



Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
Translate English to French:
cheese =>
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

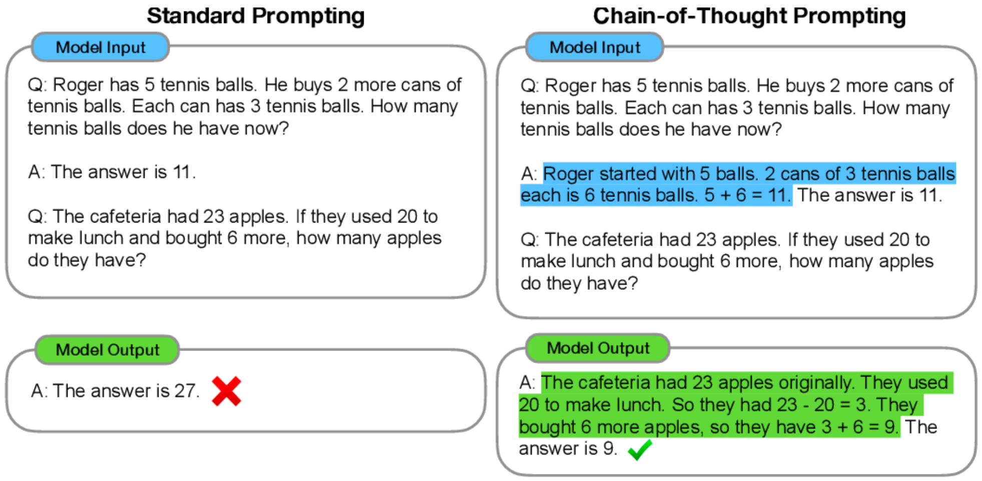
```
Translate English to French:
sea otter => loutre de mer
cheese =>
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>
```

Figure 5.3.2: (a) Zero-shot prompting, (b) One-shot prompting, (c) Few-shot prompting [11]



Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The answer is 27.

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9.

Figure 5.3.3: Chain-of-thought [84]

3. **Example Selection**: The examples with the highest cosine similarity scores are selected as the few-shot examples for prompting.

By selecting examples that are semantically close to the new task, the model is more likely to generate accurate completions or predictions. This method ensures that the selected examples are contextually relevant, thereby improving the performance of few-shot prompting without the need for manual curation.

This approach is particularly useful in tasks where examples with similar structures or semantics can help guide the model towards better performance. Furthermore, embedding-based similarity searches are scalable and efficient, making them well-suited for large datasets and complex tasks.

5.4 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) is a powerful architecture paradigm that combines the strengths of retrieval-based and generation-based approaches to enhance language model performance, particularly in knowledge-intensive tasks. Unlike standard large language models (LLMs) that rely solely on information encoded during pretraining, RAG systems augment generation with access to external knowledge sources at inference time, improving factual accuracy, scalability, and adaptability.

5.4.1 Motivation

Traditional LLMs have a fixed knowledge capacity constrained by the data seen during training and by the model's finite parameters. This limitation becomes particularly evident in domains where up-to-date, domain-specific, or rare information is critical. RAG addresses this by enabling dynamic retrieval from external document corpora or knowledge bases, thus allowing the model to "look up" relevant context before generating a response.

5.4.2 Core Architecture

A typical RAG system consists of two main components:

- **Retriever:** Given a query (e.g., a user prompt), this module retrieves relevant documents, passages, or knowledge snippets from an indexed corpus. Common retrievers include sparse methods like BM25, dense vector retrievers [37], and hybrid approaches that combine both.
- **Generator:** A generative LLM (e.g., BART, T5, or GPT) that conditions its output on both the original query and the retrieved content. The model learns to integrate retrieved evidence with language generation to produce accurate, coherent, and grounded responses.

5.4.3 Retrieval Strategies

Retrieval can be performed in various ways:

- **Single-pass retrieval:** The retriever selects a fixed number of documents before generation.
- **Multi-step or iterative retrieval:** Retrieval is performed iteratively, refining the query based on prior generations.
- **Lightweight retrieval with graph-aware structure:** Techniques like LightRAG[31] and GraphRAG[22] use document graphs and entity relationships to enhance retrieval precision and reasoning depth.

5.4.4 Advantages of RAG

RAG offers several notable advantages:

- **Improved factual accuracy:** The model grounds its responses in retrieved evidence, reducing hallucinations.
- **Scalability:** External knowledge sources can be updated independently of the model, extending longevity and adaptability.
- **Domain adaptability:** RAG is particularly effective in specialized domains (e.g., law, medicine) where pretrained knowledge may be insufficient.
- **Reduced need for fine-tuning:** Since external documents provide task-specific context, the model often requires less supervised data.

5.5 Multi-Agent Systems

Multi-Agent Systems (MAS) are computational frameworks composed of multiple interacting agents, each capable of autonomous decision-making and acting within a shared environment. These systems have gained renewed interest in the era of Large Language Models (LLMs), enabling modular, specialized, and collaborative problem-solving architectures where distinct agents can be endowed with individual roles, knowledge, or capabilities.

5.5.1 Definition and Motivation

An **agent** is a computational entity equipped with perception, reasoning, and action capabilities. It operates either independently or in coordination with others to achieve defined objectives. In a MAS, agents can collaborate, compete, negotiate, or coordinate to accomplish complex tasks that are difficult—or even impossible—for a single monolithic system to handle efficiently.

Incorporating LLMs into MAS frameworks allows for the construction of specialized language agents—each with domain-specific prompt engineering, tools, or memory—that interact to solve multi-step, multi-perspective, or high-stakes problems such as legal analysis, scientific research, or code generation.

5.5.2 Key Characteristics of Agents in MAS

Agents in a Multi-Agent System typically exhibit the following properties:

- **Autonomy:** Each agent operates without direct external intervention, making decisions based on its internal goals and knowledge.
- **Partial Observability:** Agents possess incomplete or localized knowledge about the environment or other agents, necessitating communication or coordination.
- **Inter-agent Communication:** Agents exchange information using structured protocols (e.g., messaging APIs, function calls, or natural language) to synchronize behavior and share insights.

- **Specialization:** Agents are often assigned roles (e.g., planner, reasoner, retriever) to distribute cognitive load and enhance performance via modularity.

5.5.3 Agent Roles in LLM-based Systems

Recent research explores a variety of agent types within LLM-powered multi-agent ecosystems:

- **Planner or Coordinator:** Determines overall strategy or task decomposition (e.g., breaking down a query into subgoals).
- **Retriever or Researcher:** Performs retrieval-augmented generation (RAG), accessing external sources to gather evidence.
- **Solver or Analyst:** Engages in complex reasoning, synthesis, or decision-making.
- **Critic or Validator:** Evaluates responses from other agents for consistency, correctness, or explainability.
- **Executor:** Interfaces with external APIs, tools, or environments to complete user actions.

5.5.4 Coordination Mechanisms

MAS frameworks may adopt different interaction paradigms:

- **Centralized Orchestration:** A central controller delegates tasks and integrates responses (common in pipeline-style architectures).
- **Decentralized Collaboration:** Agents operate peer-to-peer, using shared protocols to reach consensus or distribute tasks.
- **Emergent Communication:** Some systems explore self-organized communication using learned languages or adaptive natural language negotiation.

5.6 Related Work

5.6.1 Overview

Recent advances in applying LLMs and RAG to legal tasks significantly boost contract analysis, legal QA, and document review [69, 41]. Domain-specific RAG frameworks, such as Legal Query RAG, improve accuracy and relevance by combining fine-tuned legal LLMs, evaluation agents, and recursive feedback, reducing hallucinations and enhancing responses to complex queries [78]. Literature surveys highlight a sharp rise in research on LLM-driven contract review, legal research, and regulatory compliance, alongside increasing methodological sophistication and expansion into multilingual, cross-jurisdictional contexts [70].

Despite these advances, the community faces persistent challenges. A major issue is the tendency of LLMs to generate hallucinated or misleading responses, especially when lacking deep domain knowledge or when retrieval mechanisms fail to surface the most relevant legal context [78, 94, 66]. One strategy that has been investigated to mitigate this issue is fine-tuning language models on dedicated legal corpora, with the aim of deepening their understanding of legal terminology, reasoning styles, and contextual subtleties [15, 16, 33]. The evaluation of generated content remains difficult, as automated metrics often do not align with expert legal judgment, and human evaluation is costly and time-consuming [78, 65, 77]. Data availability and quality are also significant hurdles, particularly for specialized legal domains or languages with limited resources [65, 3]. Furthermore, the complexity of legal language and reasoning, as well as the need for transparency and explainability in AI-generated outputs, present ongoing obstacles [78, 3, 10]. Approaches, such as integrating case-based reasoning with RAG [88], leveraging knowledge graphs [10], and developing new retrieval and evaluation strategies [3, 66, 65], are being explored to address these limitations.

Additionally, multi-agent frameworks have recently gained attention, enhancing the reasoning capabilities and reliability of legal AI systems [68, 73, 93, 47]. Inspired by prior work such as STORM[67], which introduces collaborative multi-turn interactions among diverse agents for content generation, and ChatLaw[18], which employs a knowledge graph-enhanced multi-agent design for legal assistance, we adopt a similar approach tailored specifically for contract analysis. PAKTON is designed to address the complex reasoning demands and rigorous transparency requirements inherent in legal tasks by orchestrating agents that iteratively interrogate, retrieve, and refine information from multiple sources.

5.6.2 Key Inspirations and differentiations

STORM [67] introduces a collaborative multi-turn conversation system for writing Wikipedia-like articles. It features Wikipedia-style writers with different perspectives, along with an expert agent that answers their questions to guide content generation. While this approach inspired the interrogation structure of our system, their focus is limited to encyclopedic article writing and does not address the specific reasoning demands or legal constraints present in contract analysis. Furthermore, their method does not incorporate multiple information sources in a retrieval-augmented setting, which is critical in the legal domain.

ChatLaw [18] proposes a multi-agent legal assistant enhanced by knowledge graphs and a mixture-of-experts architecture. While its high-level structure bears similarity to our multi-agent design, it differs significantly in the implementation of retrieval-augmented techniques, agent interactions, and the process of refining answers. Moreover, its focus is primarily on Chinese legal case law rather than English-language contracts. The limited availability of open-source resources, including datasets, models, and implementation details, poses challenges for reproducibility and broader adoption across different legal systems.

Chain-of-Retrieval Augmented Generation (CoRAG) [79] presents an iterative retrieval framework that refines generation through multiple stages of retrieval. This idea of chaining retrieval steps influenced the design of our Researcher–Interrogator interaction, where iterative questioning leads to higher confidence and better-informed responses. However, CoRAG is a general framework and does not target the legal domain, nor has it been evaluated directly on legal applications.

Chapter 6

PAKTON Framework

In this chapter, we present the full architecture and operational workflow of **PAKTON**, our proposed multi-agent framework for contract analysis. The chapter outlines how PAKTON systematically addresses the unique challenges of legal document understanding by decomposing the task into distinct but interacting components. We begin by providing a high-level overview of the system's functionality and objectives, including its emphasis on transparency, retrieval grounding, and structured legal reporting. We then describe the three primary agents—*Archivist*, *Interrogator*, and *Researcher*—detailing the role and internal logic of each within the overall pipeline. The Archivist is responsible for document ingestion and hierarchical chunking, enabling fine-grained, context-aware retrieval. The Interrogator executes iterative reasoning over retrieved evidence, synthesizing user-specific legal reports with explicit justifications and gap identification. The Researcher handles in-document and cross-document retrieval using hybrid and graph-aware techniques, supplemented by external tools when applicable.

The chapter further elaborates on the underlying prompting strategies employed across all agents, particularly the use of Highlighted Chain-of-Thought (HoT) prompting to enhance robustness and clarity. Through this modular, explainable, and RAG-enhanced architecture, PAKTON achieves both performance and interpretability—two critical desiderata in high-stakes legal applications. Each section in this chapter explains the technical details, design motivations, and architectural choices that make PAKTON an effective and deployable solution for automated contract review.

6.1 Overview

PAKTON aims to analyze user-provided contracts and generate query responses grounded in the contract's content while integrating pertinent external knowledge as illustrated in Fig. 6.1.1. A key design goal is to ensure transparent and traceable reasoning by referencing evidence spans from the contract and articulating the rationale behind each conclusion. The final output is delivered as a structured legal report.

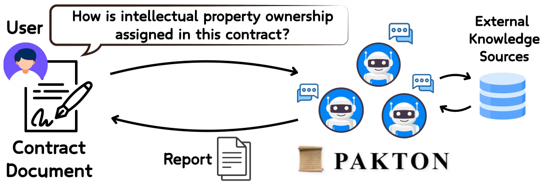

User
How is intellectual property ownership assigned in this contract?
External Knowledge Sources
Contract Document
Report
PAKTON

Figure 6.1.1: PAKTON user flow: legal query submission followed by comprehensive report generation.

To operationalize this functionality, the system employs a tri-agent architecture comprising the Archivist, Interrogator, and Researcher, each fulfilling a distinct role in the retrieval, reasoning, and synthesis pipeline. Each agent is implemented as a reasoning loop following the ReAct paradigm[91], which combines reasoning and action steps via a backbone LLM. This design enables agents to reason over their current state, generate tool-specific actions and iteratively refine their decisions based on intermediate outcomes. An overview of the framework is illustrated in Fig. 6.1.2. Further details are provided in App. A.

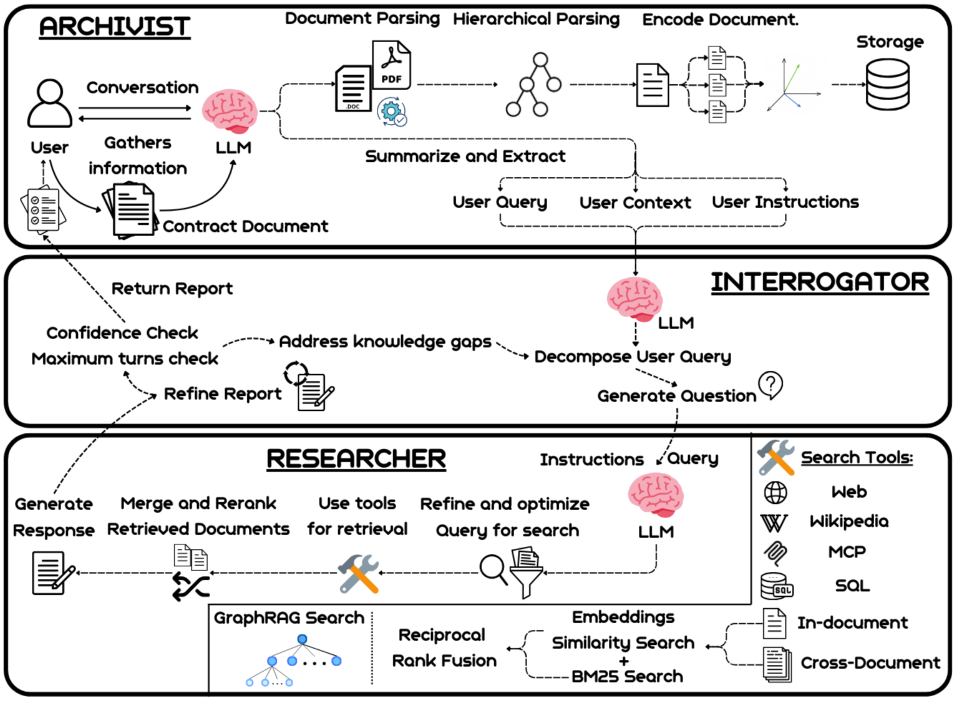


Figure 6.1.2: An overview of the proposed PAKTON framework and its internal components.

6.2 Archivist

The Archivist is responsible for gathering and organizing relevant user information. It plays a central role in the initial stages of the system by collecting and summarizing the user's query, any accompanying instructions, and contextual background. This information is then structured and passed to the Interrogator agent. The Archivist also manages storage and embedding of the user's contract document and actively engages in dialogue to resolve ambiguities or request missing user information, ultimately implementing three core functionalities:

Document Parsing. To enable practical deployment, the Archivist ingests diverse document formats within an integrated information pipeline. It natively parses text-based files, preserving structure such as headings, paragraphs, and numbered clauses. For PDFs, it applies OCR methods [71, 72] with visual layout analysis to recover document hierarchies, including sections, tables, and multi-column formats. All extracted content is then filtered and normalized to ensure consistent, structured input for downstream processing.

Hierarchical Parsing. Legal documents typically feature complex structures, cross-references, and nuanced semantic variations that render flat or linear representations inadequate for accurate analysis. To overcome this, we generate a hierarchical tree that mirrors the contract's internal organization. The contract text is parsed into this tree structure by leveraging structural cues in conjunction with either semantic similarity measures based on BERT embeddings [80] or from LLMs.

The structural parsing process involves two main steps as shown in Figure 6.2.1: (1) identifying the distinct sections of the document, and (2) determining the hierarchical relationships between those sections. Examples of sections include titles, clauses, paragraphs, and enumerated list items. A section is considered the child of another if, structurally, it appears nested within the parent section based on indentation, numbering patterns, visual layout (in OCR-extracted documents), or semantic proximity.

Encode Document. The tree representation of the contract is used to create contextualized chunks that are then embedded for retrieval. Context-aware chunking has been shown to reduce failed retrievals by preserving semantic coherence across sections[4]. In particular, both overlapping[81] and dynamic chunking[21] have demonstrated gains in retrieval performance by maintaining relevant context around each chunk boundary. To capture different contextual information levels, we generate three chunk types for each tree node, which are later concatenated into a unified chunk set. Duplicate or highly similar chunks are filtered out to improve embedding diversity and avoid redundancy. The chunking strategies are as follows:

- **Node-level:** Encodes each node in isolation to preserve fine-grained semantics. This supports precise alignment with user queries targeting specific clauses or legal provisions and ensures that each atomic unit is independently retrievable.
- **Ancestor-aware:** Concatenates a node with its hierarchical ancestors, capturing inherited context from section headers, articles, and titles. This contextualization aids disambiguation and enhances understanding of a section's role within the broader contractual structure.
- **Descendant-aware:** Aggregates a node with its descendants to embed cohesive semantic units, such as clauses with their subclauses or enumerations. This is particularly beneficial for reasoning over compound or multi-part provisions, where meaning is distributed across nested content.

Each chunk is further enriched with metadata, including its structural location within the tree, document position, filename, and a contract-level summary. This metadata facilitates both cross-document retrieval—by guiding vector store routing—and intra-document relevance ranking. By integrating multiple structural perspectives, our chunking framework improves the likelihood of retrieving relevant content across a spectrum of query granularities, from clause-specific lookups to section-level reasoning.

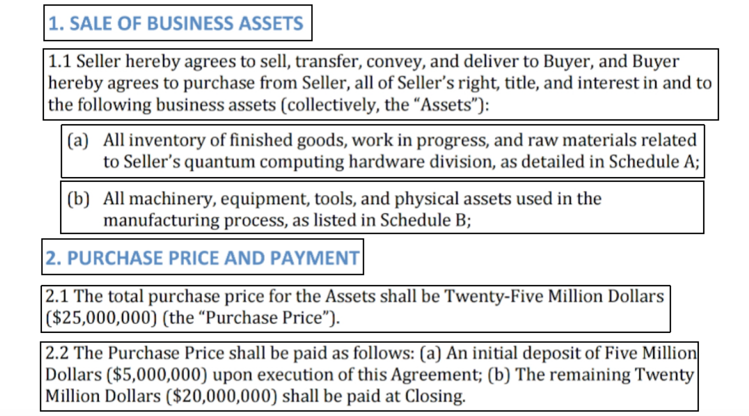
6.3 Interrogator

The *Interrogator* is responsible for generating the final report presented to the user as the system's response. To this end, it orchestrates a multi-step reasoning process aimed at addressing queries with both accuracy and confidence. It takes as input the user's query, contextual information, and instructions, and initiates an iterative interrogation of the Researcher agent. This process involves decomposing the original query and generating a series of questions aimed at refining and deepening the system's understanding. Once a response is received from the Researcher for the first question, the Interrogator generates a preliminary report that attempts to directly answer the user's query. This initial response also serves to identify potential knowledge gaps that require further research.

Based on this initial exchange, the Interrogator formulates follow-up questions aimed at clarifying missing information in the preliminary answer. When generating each subsequent question, it considers the user query, context, instructions, a summary of previously asked questions, and the current draft of the report. Questions that are most likely to significantly improve the final response are prioritized. With each new response from the Researcher, the Interrogator incrementally refines the draft report. The process continues until one of two stopping conditions is met: 1) the Interrogator determines that the answer is sufficiently confident and complete, or 2) the maximum number of allowed interrogation turns (simulating retrieval depth), set by the user, has been reached.

The Interrogator is explicitly prompted to avoid speculation and to stay grounded in the retrieved content and to clearly highlight any ambiguities or missing information that prevent full resolution of the query. Additionally, the Interrogator is responsible for validating the structure of the final report, which must include:

- A Title and a concise topic Summary.
  EchoPrompt[53] has shown that prompting the model to rephrase the question during the generation phase, before producing an answer, enhances in-context learning performance. This applies when the Interrogator drafts the initial report or revises an existing one, incorporating the title and summary into the output before providing any conclusions or additional information.
  Additionally, Re-reading[90] has demonstrated that repeating the exact question twice in the input improves the model's answer accuracy. This becomes relevant when the Interrogator refines the draft report, which includes the title and summary alongside the original query in the prompt, ensuring alignment



1. SALE OF BUSINESS ASSETS

1.1 Seller hereby agrees to sell, transfer, convey, and deliver to Buyer, and Buyer hereby agrees to purchase from Seller, all of Seller's right, title, and interest in and to the following business assets (collectively, the "Assets"):

(a) All inventory of finished goods, work in progress, and raw materials related to Seller's quantum computing hardware division, as detailed in Schedule A;

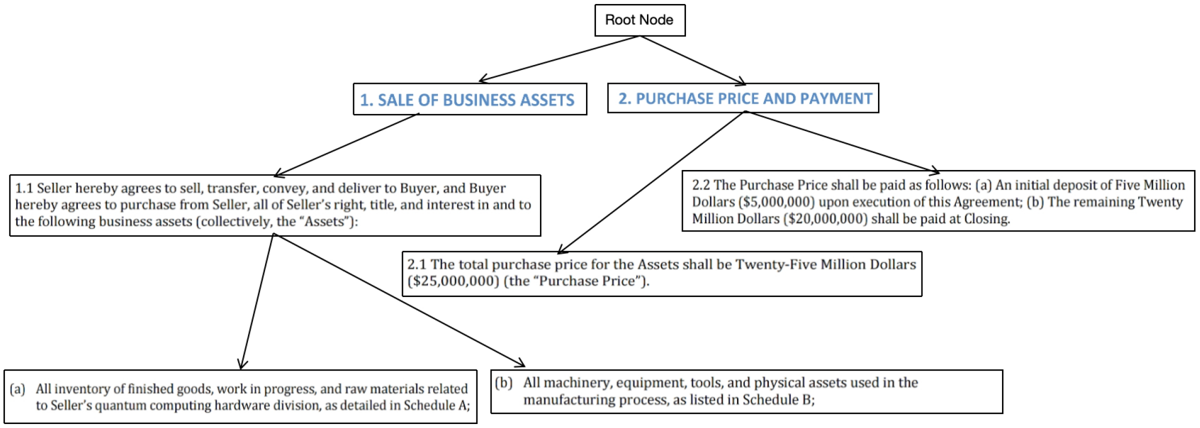
(b) All machinery, equipment, tools, and physical assets used in the manufacturing process, as listed in Schedule B;

2. PURCHASE PRICE AND PAYMENT

2.1 The total purchase price for the Assets shall be Twenty-Five Million Dollars ($25,000,000) (the "Purchase Price").

2.2 The Purchase Price shall be paid as follows: (a) An initial deposit of Five Million Dollars ($5,000,000) upon execution of this Agreement; (b) The remaining Twenty Million Dollars ($20,000,000) shall be paid at Closing.

(a) Section Detection



1.1 Seller hereby agrees to sell, transfer, convey, and deliver to Buyer, and Buyer hereby agrees to purchase from Seller, all of Seller's right, title, and interest in and to the following business assets (collectively, the "Assets"):

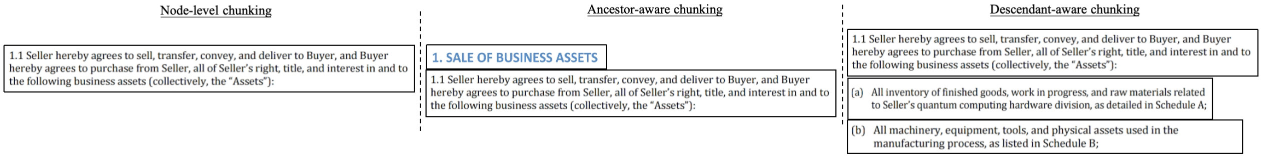
2.2 The Purchase Price shall be paid as follows: (a) An initial deposit of Five Million Dollars ($5,000,000) upon execution of this Agreement; (b) The remaining Twenty Million Dollars ($20,000,000) shall be paid at Closing.

2.1 The total purchase price for the Assets shall be Twenty-Five Million Dollars ($25,000,000) (the "Purchase Price").

(a) All inventory of finished goods, work in progress, and raw materials related to Seller's quantum computing hardware division, as detailed in Schedule A;

(b) All machinery, equipment, tools, and physical assets used in the manufacturing process, as listed in Schedule B;

(b) Hierarchical Organization of sections.



1.1 Seller hereby agrees to sell, transfer, convey, and deliver to Buyer, and Buyer hereby agrees to purchase from Seller, all of Seller's right, title, and interest in and to the following business assets (collectively, the "Assets"):

1.1 Seller hereby agrees to sell, transfer, convey, and deliver to Buyer, and Buyer hereby agrees to purchase from Seller, all of Seller's right, title, and interest in and to the following business assets (collectively, the "Assets"):

1.1 Seller hereby agrees to sell, transfer, convey, and deliver to Buyer, and Buyer hereby agrees to purchase from Seller, all of Seller's right, title, and interest in and to the following business assets (collectively, the "Assets"):

(a) All inventory of finished goods, work in progress, and raw materials related to Seller's quantum computing hardware division, as detailed in Schedule A;

(b) All machinery, equipment, tools, and physical assets used in the manufacturing process, as listed in Schedule B;

(c) Contextual Embeddings for node "1.1 Seller ..."

Figure 6.2.1: Tree Representation and Chunking Workflow for Contract Documents

with the topic.
[53, 90] have shown that prompting strategies such as question rephrasing and repetition significantly enhance in-context learning and response accuracy. Incorporating a clear title and summary at each refinement round at the start of the report operationalizes these strategies by reinforcing topic focus and aligning the model's generation with the user's intent. This also facilitates early detection of misinterpretations by making the system's understanding of the query explicit. As a result, including a title and summary not only improves user readability but also enhances overall model performance.

- Legal reasoning and key findings, enhancing transparency
- A preliminary answer and direction for further research
- Identified knowledge gaps and proposed next questions
- A list of supporting sources

The iterative architecture of the *Interrogator* further facilitates the identification of knowledge gaps, targeted information retrieval, and progressive query disambiguation. Notably, the inherent repetition in this process has been shown to enhance in-context learning and output accuracy[90]. These design choices are integral to the overall effectiveness of PAKTON.

We also explored an alternative approach in which the entire interrogation history was provided to the LLM to generate the final report in a single step. However, this method proved less effective in practice, as relevant information was often lost or diluted in the middle of the context[46]. Moreover, smaller models struggled to manage the extended input, leading to reduced performance and difficulty in accurately locating supporting evidence.

6.4 Researcher

The *Researcher* is responsible for retrieving relevant information to support the Interrogator in answering the user's query. As an agent, it is equipped with multiple retrieval methods, each encapsulated as a tool. Depending on the nature of the query and the accompanying instructions, the Researcher autonomously selects the most suitable retrieval method, or combination of methods. This selection process is driven by prompting an LLM to choose the optimal set of tools based on the query content and tool descriptions. Retrieval methods are categorized into two primary types:

- **In-document retrieval**: Leveraging the document chunks and embeddings provided by the *Archivist*, the Researcher retrieves spans to address the query. A hybrid retrieval approach is employed, combining BM25[64], dense embeddings[44], and Reciprocal Rank Fusion (RRF)[17]. This is further enhanced by LightRAG[31] (a lightweight version of GraphRAG [22]), improving entity- and relation-level matching within and across documents, and is particularly effective for answering global queries that require reasoning over multiple interrelated documents.
- **Cross-document retrieval**: This method retrieves relevant spans from other documents, which the Interrogator can leverage as exemplars in few-shot prompting.
  PAKTON supports additional retrieval capabilities, including web search, Wikipedia access, SQL database querying, and integration with external legal databases via the Model Context Protocol (MCP)[5]. These tools are designed to supplement the model's knowledge in cases where it lacks sufficient information or requires real-time updates. However, to ensure a fair and controlled evaluation, this category of retrieval was not employed in our experiments.

To ensure both high recall and precision, the Researcher employs a two-step retrieval and reranking process. Initially, high-recall retrievers, such as the aforementioned, are used to collect a broad set of potentially relevant passages maximizing coverage. Subsequently, a cross-encoder model reranks the retrieved results by jointly encoding the query and each passage, allowing for more accurate semantic relevance estimation[37]. This reranking stage serves as a high-precision filter that refines the initial candidate set. Relevance scores are normalized using a sigmoid function, and passages exceeding a predefined similarity threshold are passed to the response generation module, which synthesizes the final answer.

6.5 Prompt Engineering

All prompts in our setup employed the Highlighted Chain-of-Thought (HoT) approach[55], wherein XML-style tags (e.g., *<context>…</context>*) are used to clearly delineate prompt components such as the user query, instructions, and retrieved context. This structured formatting has been shown to enhance model reasoning by increasing the salience of key information[55], and to improve robustness against prompt injection by helping models distinguish between functional sections of the prompt[45].

Chapter 7

Experiments and Results

In this chapter, the experiments and results are presented to assess the effectiveness of the proposed framework across multiple evaluation dimensions. PAKTON is evaluated through both quantitative and qualitative methods. In the quantitative analysis, it is tested on a contract analysis task, where it outperforms all baselines in terms of accuracy. To isolate component-level contributions, the Archivist–Researcher modules—responsible for RAG—are evaluated on four benchmark datasets, achieving state-of-the-art results. For the Interrogator—responsible for generating the user-facing answer—a qualitative evaluation is conducted, with outputs assessed against nine criteria aligned with the system's goal of producing interpretable and actionable responses. This includes a structured human evaluation by five expert attorneys and a Supreme Court Justice, alongside an automated assessment using G-EVAL [49]. In both evaluations, PAKTON consistently outperforms GPT-4o, exceeding it in eight of the nine criteria.

7.1 Quantitative Results

Quantitative evaluation provides a rigorous, objective foundation for assessing the effectiveness and robustness of the PAKTON framework. In this section, we present empirical results from benchmark datasets that quantitatively measure both classification accuracy and retrieval performance. These metrics allow us to isolate and compare the contributions of individual system components, such as the reasoning and retrieval modules, under controlled conditions. By benchmarking against strong baselines—including fine-tuned legal models and advanced RAG techniques—we demonstrate that PAKTON not only achieves superior predictive performance, but also exhibits remarkable robustness across varying language models. These findings offer concrete evidence of the framework's generalizability, effectiveness, and practical applicability to real-world contract analysis tasks.

7.1.1 Performance on a classification dataset

Setup. For the quantitative evaluation, PAKTON was evaluated on the ContractNLI dataset [38]. In this dataset, a premise denotes an entire contract document, with each premise paired with a corresponding hypothesis. The classification task involves determining whether the hypothesis is entailed by, contradicted by, or not addressed in (neutral with respect to) the associated contract. Given the substantial length of the contracts, the dataset also provides annotated spans that indicate the specific portions of text necessary for making the classification decision.

We evaluate PAKTON on ContractNLI by comparing its performance against several baselines, including models specifically pretrained on legal corpora (e.g., Saul[15]) and models employing different prompting techniques (Table 7.1). The evaluation metrics presented include overall accuracy and the weighted F1-score (F1[W]), alongside the individual F1-scores for the entailment, contradiction, and neutral classes.

The prompting approaches examined include zero-shot (ZS), few-shot (FS)—where entire contractual documents are used as exemplars—and an alternative few-shot setting (FS-Spans), in which only the relevant spans influencing classification are provided. Refer to App. C for a detailed description of the experimental setup. The results indicate a clear superiority of PAKTON across all evaluated methods, notably outperforming even domain-specific fine-tuned models.

Results. Firstly, PAKTON consistently outperforms the baseline methods across all models. Comparing the results of Mistral and Mixtral with Saul, we observe that PAKTON yields better performance even compared to fine-tuning on related legal corpora. Notably, Mistral 7B with PAKTON surpasses Saul 54B—a model derived from the larger Mixtral 54B and fine-tuned on legal data. Furthermore, Mistral 7B with PAKTON even outperforms GPT-4o in a FS setting.

Furthermore, a notable observation emerges from comparing the performance across different core language models utilizing our framework. The variation in performance among models integrated with PAKTON was minimal, highlighting the robustness of the proposed framework. For example, the F1[W] for Llama 3.1 70B score with PAKTON is 79.03%, while for Gemma 3 27B is 82.83%, yielding a modest difference of only 3.8 percentage points. In contrast, the performance gap between these models under the ZS prompting scenario is significantly larger (22.83%), clearly demonstrating that Gemma 3 27B substantially outperforms Llama 3.1 70B in the baseline ZS setting. This considerable reduction in performance disparity indicates that PAKTON's architecture effectively mitigates the inherent variability among underlying LLMs, ensuring consistently high performance regardless of the core model employed.

To further investigate PAKTON's robustness, we conduct a targeted statistical analysis focusing on variability and dependency. First, we compute the coefficient of variation (CV) across all models' PAKTON–based F1[W] scores, obtaining a low CV of 12.6% (for comparison, ZS scores typically exhibit CVs > 25%). This indicates minimal relative variability and, consequently, high robustness. Second, we perform a one–way ANOVA by splitting models into two groups according to the median of their ZS scores. The test reveals no statistically significant difference in PAKTON performance between the high– and low–ZS groups ($F - statistic = 3.05$, $p = 0.12$). Lastly, linear regression analysis is conducted to assess the direct relationship between baseline ZS performance and PAKTON, yielding a shallow slope of 0.44, demonstrating that every unit of baseline gain translates into less than half a unit under PAKTON, thereby compressing absolute performance gaps.

Collectively, these findings demonstrate that PAKTON substantially reduces performance disparities among diverse LLMs while preserving consistently high accuracy. Such robustness is particularly advantageous in the legal domain, where relying on open–source models alleviates the privacy risks associated with sending sensitive contractual or legal information to proprietary systems.

7.1.2 Performance of RAG

Setup. In addition to ContractNLI, we leverage the LegalBench-RAG benchmark [60] to assess the retrieval pipeline—specifically, the Archivist and Researcher modules—independently of answer generation. This isolation is essential, as PAKTON's overall performance hinges on the relevance of retrieved context. LegalBench-RAG covers four contract-related domains—NDAs, M&A agreements, commercial contracts, and consumer-facing privacy policies—enabling a robust evaluation of retrieval accuracy across heterogeneous legal corpora. We benchmark PAKTON's RAG module against four baselines used in [60], namely: Naive (fixed 500-character chunks with OpenAI embeddings[56]), RCTS (structure-aware splitting[42]), Naive+Cohere (chunking with Cohere reranking[1]), and RCTS+Cohere (structural chunking with reranking). Full details are provided in App. B.

Results. Table 7.2 presents a comparative analysis of retrieval performance across diverse legal corpora, using precision and recall at various k thresholds to benchmark PAKTON against established baselines. Across all datasets, PAKTON consistently and significantly outperforms alternatives at every evaluated k. For instance, on the ContractNLI dataset, PAKTON achieves a Recall@1 of 53.14%, nearly 5 times higher than the strongest baseline (11.32%), a trend that holds across all k values. Similar improvements are observed in other datasets: on PrivacyQA, MAUD, and CUAD, PAKTON surpasses the best Recall@1 scores by margins often exceeding 20%. Aggregate results reinforce this pattern, with PAKTON achieving more than five-fold increase in Recall@1 (26.77% vs. 4.94%). These improvements are especially critical in the legal domain, where high recall is essential. Failing to retrieve relevants spans can result in flawed reasoning

---

[1] rerank-english-v3.0

or unsupported conclusions, particularly when legal documents contain conflicting clauses, exceptions, or interdependent provisions that must be interpreted in context.

7.2 Qualitative Results

Unlike quantitative metrics, which focus on measurable aspects such as accuracy or retrieval precision, the qualitative evaluation centers on the perceived quality and interpretability of the generated responses. This assessment reflects how well the system supports real-world decision-making, particularly in legal contexts where clarity, justification, and practical utility are paramount. In essence, these results capture how end users—especially non-experts—interpret and value the system's outputs, offering insights into the effectiveness of the responses beyond what numerical scores alone can reveal. Comparison of PAKTON outputs and GPT-4o can be seen in App. F.

Setup. To evaluate PAKTON's practical utility, we conduct a human study using 15 benchmark legal questions crafted by five attorneys and a Supreme Court Justice to capture authentic legal reasoning. In a *Prolific*[2] survey, participants compare paired responses from PAKTON and ChatGPT[3] across nine criteria: explainability, evidence, legal/contextual understanding, ambiguity handling, gap acknowledgment, conciseness, coherence, relevance, and completeness. For each, they select the best answer and provided justification, with "None" and "Not Sure" options to ensure robustness. ChatGPT serves as a baseline given its adoption and, for fairness, leverage RAG when external documents were provided [57]. Additional methodological details are in App. D.

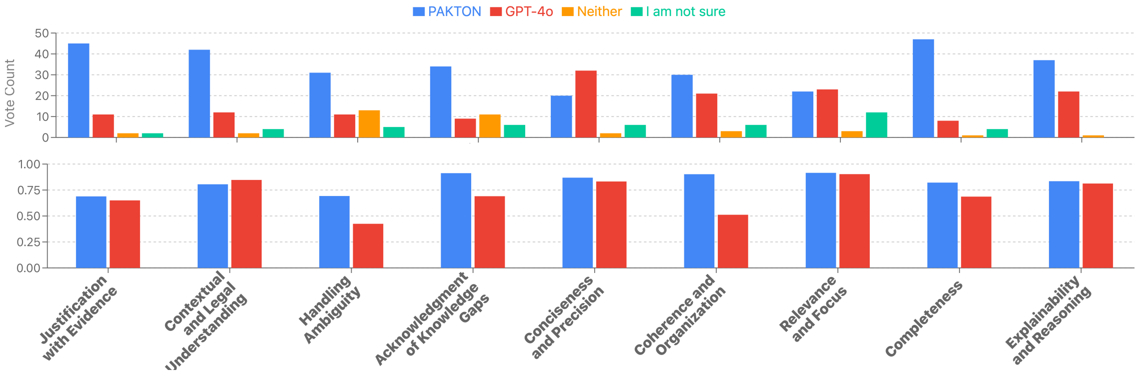


Figure 7.2.1: Comparison analysis of PAKTON and GPT-4o. Top plot presents human preferences across nine evaluation criteria aggregated for all questions. Bottom plot shows G-EVAL scores for the same criteria, aggregated across all ContractNLI outputs.

Results. Sixty participants completed the evaluation, comparing PAKTON and ChatGPT outputs across 9 criteria for 15 legal questions, yielding 540 individual judgments. As illustrated in Figure 7.2.1, PAKTON is favored over ChatGPT on the majority of evaluation dimensions, a trend consistent across all legal scenarios assessed. The most pronounced advantage is observed in "Completeness," where PAKTON consistently provides more comprehensive responses. PAKTON also excels in "Explainability and Reasoning," offering clearer, stepwise rationales—an important attribute for non-specialist users. To assess the participants' discriminative capacity, we include competing criteria such as "Completeness" versus "Conciseness and Precision," recognizing the inherent tension between exhaustive and succinct responses; as expected, PAKTON excells in completeness, while ChatGPT is preferred for conciseness. For "Relevance and Focus," both models receive comparable scores, reflecting a trade-off between breadth and specificity. These outcomes confirm PAKTON's core desiderata to generate detailed, report-like answers, with the observed completeness-precision trade-off both anticipated and justified.

---

[2] https://www.prolific.com

[3] gpt-4o-2024-11-20

7.3 Evaluation using LLMs

The quality of each system's responses is further evaluated using a large language model (LLM), as detailed in the following section.

Setup. To comprehensively assess PAKTON at scale, we employed G-EVAL[49], a widely used LLM-as-a-Judge evaluation framework[51, 82, 14] for NLG output quality across the same criteria as the human study. We evaluated 102 randomly selected samples from ContractNLI[38], comparing PAKTON and GPT-4o on matched inputs. G-EVAL generated criterion-specific scores, supporting fine-grained analysis of response quality, explainability, and reasoning.

Results across all samples and evaluation criteria prove PAKTON's superiority over GPT-4o in eight out of nine dimensions (Fig. 7.2.1). Notably, PAKTON scores higher in Explainability and Reasoning, Justification with Evidence, Completeness, and Handling Ambiguity. These outcomes are consistent with PAKTON's design objectives that prioritize detailed and well-supported responses that explicitly reason through legal content. For Conciseness and Precision, and Relevance and Focus criteria, both models demonstrate comparable performance, an anticipated outcome, as these dimensions often conflict with Completeness—a domain in which PAKTON significantly surpasses GPT-4o—thereby illustrating the inherent trade-off between brevity and depth. The sole criterion in which PAKTON underperforms relative to GPT-4o is Contextual and Legal Understanding; while this initially appears counterintuitive, a closer examination of the G-EVAL rationale reveals that PAKTON's responses frequently acknowledge knowledge gaps under uncertainty. Although this is desirable from a transparency perspective, the evaluation framework interprets such acknowledgments as evidence of limited understanding, resulting in lower scores for this dimension (App. E).

Model	**Method**	**Acc.**	**F1 [W]**	**F1 [E]**	**F1 [C]**	**F1 [N]**
Saul7B	Inst. Tun.	0.4196	0.2900	0.0589	0.0680	0.5920
Saul54B	Inst. Tun.	0.7020	0.6792	0.7727	0.1729	0.7024
Mistral 7B	ZS	0.5364	0.5042	0.5279	0.0248	0.5951
	FS	0.5065	0.4702	0.6053	0.0082	0.4379
	FS+Spans	0.4940	0.4576	0.6085	0.0076	0.4053
	PAKTON	0.7032	0.6789	0.7782	0.2469	0.6828
Mixtral 8x7B	ZS	0.5423	0.5475	0.6445	0.4103	0.4770
	FS	0.6002	0.5804	0.6836	0.1931	0.5642
	FS+Spans	0.6150	0.6017	0.6901	0.1951	0.6060
	PAKTON	0.7423	0.7429	0.7864	0.6655	0.7187
Qwen 2.5 72B	ZS	0.7728	0.7699	0.8248	0.5776	0.7579
	FS	0.7351	0.7241	0.8094	0.4920	0.6892
	FS+Spans	0.7484	0.7432	0.8196	0.4378	0.7357
	PAKTON	**0.8192**	**0.8188**	0.8353	0.7737	0.8132
Gemma 3 27B	ZS	0.7886	0.7860	0.8316	0.6348	0.7739
	FS	0.7191	0.7049	0.7815	0.4608	0.6891
	FS+Spans	0.7720	0.7639	0.8287	0.4728	0.7662
	PAKTON	**0.8287**	**0.8283**	0.8487	0.7546	0.8255
Llama 3.3 70B	ZS	0.6767	0.6716	0.7366	0.5378	0.6346
	FS	0.6657	0.6565	0.7326	0.4431	0.6268
	FS+Spans	0.6915	0.6879	0.7382	0.4244	0.6982
	PAKTON	**0.8217**	**0.8207**	0.8422	0.7488	0.8165
Llama 3.1 70B	ZS	0.5811	0.5577	0.5216	0.3152	0.6555
	FS	0.5729	0.5506	0.5421	0.2381	0.6358
	FS+Spans	0.5538	0.5180	0.4471	0.3014	0.6468
	PAKTON	0.7916	0.7903	0.8097	0.6846	0.7960
Claude 3.5	ZS	0.7916	0.7977	0.8757	0.5722	0.7691
	FS	0.7778	0.7816	0.8588	0.5702	0.7505
	FS+Spans	0.7999	0.8034	0.8678	0.6046	0.7826
	PAKTON	0.7990	0.8000	0.8157	0.7046	0.8072
Claude 3.7	ZS	0.7704	0.7781	0.8633	0.5602	0.7398
	FS	0.7590	0.7602	0.8463	0.5607	0.7165
	FS+Spans	0.7724	0.7766	0.8538	0.5805	0.7417
	PAKTON	0.8247	0.8254	0.8386	0.7495	0.8304
Deepseek V3	ZS	0.7886	0.7875	0.8487	0.6117	0.7648
	FS	0.7681	0.7607	0.8346	0.6104	0.7182
	FS+Spans	0.7743	0.7714	0.8377	0.5812	0.7465
	PAKTON	0.8192	0.8200	0.8315	0.7615	0.8224
GPT-4o	ZS	0.6121	0.6366	0.7490	0.4162	0.5698
	FS	0.6640	0.6789	0.7372	0.4734	0.6666
	FS+Spans	0.6482	0.6574	0.6664	0.4636	0.6950
	PAKTON	0.7966	0.7972	0.7964	0.7592	0.8068

Table 7.1: Performance comparison of PAKTON and other methods across models on the ContractNLI test set. The highest accuracy and F1[w] are shown in **bold**.

		Precision @ k							**Recall @ k**						
Dataset	**Method**	1	2	4	8	16	32	64	1	2	4	8	16	32	64
PrivacyQA	Naive	7.86	7.31	6.41	5.06	3.58	2.41	1.54	7.45	12.53	20.88	32.38	42.45	54.27	66.07
	RCTS	14.38	13.55	12.34	9.03	6.06	4.17	2.81	8.85	15.21	27.92	42.37	55.12	71.19	84.19
	Naive + Cohere	14.38	13.55	12.34	9.02	6.06	4.17	2.81	8.85	15.21	27.92	42.37	55.12	71.19	84.19
	RCTS + Cohere	13.94	15.91	13.32	9.57	6.88	4.68	3.28	7.32	16.12	25.65	35.60	51.87	64.98	79.61
	PAKTON	**19.94**	**16.84**	**11.44**	**8.62**	**7.38**	**6.42**	**6.08**	**13.34**	**22.43**	**32.67**	**43.39**	**61.65**	**82.30**	**89.42**
ContractNLI	Naive	16.45	14.80	12.53	9.73	6.70	4.65	3.04	11.32	19.10	29.79	45.59	56.75	69.88	86.57
	RCTS	6.63	5.29	3.89	2.81	1.98	1.29	0.90	7.63	11.33	17.34	24.99	35.80	46.57	61.72
	Naive + Cohere	6.63	5.28	3.89	2.81	1.98	1.29	0.90	7.63	11.34	17.34	24.99	35.80	46.57	61.72
	RCTS + Cohere	5.08	5.59	5.04	3.67	2.52	1.75	1.17	4.91	9.33	16.09	25.83	35.04	46.90	62.97
	PAKTON	**33.02**	**30.34**	**17.33**	**9.98**	**5.87**	**4.68**	**4.52**	**53.14**	**67.47**	**80.06**	**89.71**	**95.50**	**99.56**	**99.82**
MAUD	Naive	3.36	2.65	2.18	1.89	1.48	1.06	0.75	2.54	3.12	4.53	8.75	13.16	18.36	25.62
	RCTS	2.65	1.77	1.96	1.40	1.39	1.15	0.82	1.65	2.09	4.59	6.18	12.93	21.04	28.28
	Naive + Cohere	2.64	1.77	1.96	1.40	1.38	1.15	0.82	1.65	2.09	5.59	6.18	12.93	21.04	28.28
	RCTS + Cohere	1.94	2.63	2.05	1.77	1.79	1.55	1.12	0.52	2.48	4.39	7.24	14.03	22.60	31.46
	PAKTON	**25.47**	**17.45**	**10.51**	**7.24**	**5.08**	**3.18**	**1.85**	**23.99**	**30.09**	**34.49**	**46.42**	**59.74**	**74.96**	**82.80**
CUAD	Naive	9.27	8.05	5.98	4.33	2.77	1.77	1.09	12.60	19.47	27.92	40.70	51.02	64.38	75.71
	RCTS	1.97	4.03	4.83	4.20	2.94	1.99	1.25	1.62	8.11	17.72	31.68	44.38	60.04	74.70
	Naive + Cohere	1.97	4.03	4.83	4.20	2.94	1.99	1.25	1.62	8.11	17.72	31.68	44.38	60.04	74.70
	RCTS + Cohere	3.53	4.18	6.18	5.06	3.93	2.74	1.66	3.17	7.33	18.26	28.67	42.50	55.66	70.19
	PAKTON	**11.02**	**8.83**	**6.81**	**4.72**	**2.78**	**2.07**	**1.62**	**16.52**	**24.76**	**33.34**	**46.67**	**59.53**	**77.08**	**86.23**
ALL	Naive	2.40	3.76	4.97	4.33	3.39	2.17	1.29	3.37	8.44	21.30	34.51	48.88	64.47	76.39
	RCTS	6.41	6.16	5.76	4.36	3.09	2.15	1.45	4.94	9.19	16.90	26.30	37.06	49.71	62.22
	Naive + Cohere	6.41	6.16	5.76	4.36	3.09	2.15	1.45	4.94	9.19	16.90	26.30	37.05	49.71	62.22
	RCTS + Cohere	6.13	7.08	6.65	5.02	3.78	2.68	1.81	3.98	8.82	16.10	24.34	35.86	47.54	61.06
	PAKTON	**22.34**	**18.37**	**11.52**	**7.63**	**5.26**	**4.08**	**3.52**	**26.77**	**36.32**	**45.26**	**56.66**	**69.17**	**83.50**	**89.58**

Table 7.2: Precision and Recall @ $k \in \{1, 2, 4, 8, 16, 32, 64\}$ for four retrieval pipelines on five legal-text datasets.

Chapter 8

Conclusion

This work presents **PAKTON**, a modular, open-source, multi-agent framework designed to meet the specific demands of contract analysis—an area that presents substantial challenges due to its inherent ambiguity, domain complexity, and the need for precision, transparency, and explainability. In contrast to black-box legal AI systems, PAKTON adopts a structured and interpretable architecture built around three collaborative agents: the *Archivist* for preprocessing and input management, the *Researcher* for hybrid, graph-aware retrieval, and the *Interrogator* for iterative multi-step reasoning. Together, these components orchestrate a retrieval-augmented generation (RAG) pipeline that produces highly contextualized, evidence-backed legal analyses and detailed report-like responses.

The framework is distinguished not only by its design principles—modularity, transparency, and adaptability—but also by its practical effectiveness. Across a series of rigorous quantitative and qualitative evaluations, PAKTON demonstrates consistent superiority over both general-purpose and domain-specialized baselines. On benchmark datasets such as ContractNLI and LegalBench-RAG, the system achieves state-of-the-art retrieval and classification performance, highlighting the strength of its retrieval pipeline. Human evaluations conducted by legal professionals, as well as LLM-based assessments using G-EVAL, further validate PAKTON's ability to generate responses that are more complete, better justified, and more explainable than those produced by even leading proprietary models like GPT-4o.

Importantly, PAKTON introduces a new level of robustness and consistency in performance across different underlying LLMs. By mitigating model variance and enabling high performance with lighter-weight, open-source models, PAKTON facilitates on-premise deployment in data-sensitive environments where privacy, security, and transparency are paramount. The observed reduction in performance disparity across models also underscores PAKTON's potential as a generalizable legal reasoning framework, agnostic to specific model architectures or sizes.

Our experiments also highlight the advantages of multi-agent orchestration for complex legal reasoning tasks. The collaborative workflow ensures that retrieval, interpretation, and synthesis are handled as distinct but interconnected responsibilities, reducing hallucinations, surfacing knowledge gaps, and enhancing accountability. In a legal domain where incomplete or misleading outputs can have significant consequences, PAKTON's structured approach ensures outputs are both grounded and explicitly traceable to contract clauses or external legal sources.

In sum, PAKTON lays the groundwork for a new generation of transparent, explainable, and practically deployable legal AI systems. By marrying open-source accessibility with agent-based modularity and rigorous evaluation, it sets a new standard for contract analysis and points toward broader applications of multi-agent RAG architectures in legal and other high-stakes domains.

Chapter 9

Limitations and Ethical Considerations

In this chapter, we critically examine the limitations of the PAKTON framework and reflect on the broader ethical considerations surrounding its use. We identify technical constraints related to generalizability, efficiency, and structural assumptions, as well as practical challenges tied to deployment latency and language coverage. Additionally, we discuss ethical risks such as overreliance by non-experts, potential misuse, and issues of accessibility and fairness. These reflections aim to contextualize the system's capabilities within real-world settings and underscore the importance of responsible development and deployment.

9.1 Limitations

While PAKTON demonstrates strong capabilities in contract analysis, it has several limitations that should be considered when interpreting its results and deploying it in real-world scenarios.

Language Scope. Our system has been tested only on English-language contracts. As legal language varies significantly across languages and cultures, additional adaptation and evaluation would be necessary for multilingual or cross-lingual applications.

Contract and Jurisdiction Coverage. PAKTON has been evaluated on a subset of contract types and does not currently cover the full diversity of legal documents. Similarly, the system has not been tested across different legal jurisdictions, each of which may have unique terminology, structure, and interpretive norms. These factors may affect the system's generalizability and legal relevance in broader contexts.

Latency and Cost. Given our focus on the quality and depth of the generated reports, the system prioritizes multi-step reasoning over speed. As a result, response times may be longer compared to general-purpose language models, particularly due to the iterative communication between agents. This design also increases computational cost, making it less suitable for low-latency or resource-constrained environments.

Explainability vs. Efficiency Tradeoff. Our framework is explicitly designed to enhance transparency and reasoning. However, this emphasis on explainability can sometimes result in longer or less concise responses. In prioritizing clarity and justification, the system may occasionally sacrifice brevity or even slightly reduce precision, especially in cases where ambiguity is high and reasoning chains are extended.

Structural Parsing Generalization. The system's structural parsing component is optimized for standard contract formats, which follow consistent patterns of headings, clauses, and subclauses. When documents deviate significantly from these conventions or lack a clearly defined structure, the benefits of structural parsing are reduced. In such cases, the parsing mechanism defaults to a more general-purpose chunking strategy. This fallback does not hinder the system's functionality but may limit the advantages gained from fine-grained hierarchical representation.

9.2 Ethical Considerations

The deployment of AI systems in legal contexts raises significant ethical questions that extend beyond technical performance. In this section, we discuss the ethical implications of using PAKTON for contract analysis, focusing on the risks of user overreliance, accessibility disparities, potential misuse, and evaluation transparency. These considerations are essential to ensure that the system is not only effective but also aligned with principles of fairness, accountability, and responsible AI deployment.

Legal Expertise and Overreliance. Our system is developed to aid in contract analysis and increase access to legal information, but it does not serve as a substitute for qualified legal advice. There is a risk that users, particularly non-experts, may over-rely on its outputs without proper legal verification. To mitigate this, we recommend clearly communicating the system's limitations and encouraging users to consult legal professionals when making important decisions. **PAKTON should be viewed as an assistive tool, not a definitive authority on legal interpretation**.

Accessibility. Contract analysis is often inaccessible to non-professionals due to its complexity and reliance on legal expertise (as discussed previously). PAKTON's goal is to help democratize contract understanding by providing explainable, user-friendly outputs that can assist individuals without legal backgrounds. We are also trying to offer free access to a publicly deployed version of the system, to the extent that it remains practically and financially feasible. Nevertheless, disparities in access to computational resources may limit the ability of low-resourced groups to deploy or benefit from the system effectively. Future iterations should explore lightweight deployments support to enhance accessibility across a broader range of users.

Security concerns and misuse. As with any system built on large language models, PAKTON may be susceptible to potential misuse and adversarial attacks (e.g. prompt injection). Malicious users might attempt to exploit the system to bypass safeguards, distort outputs, or extract sensitive information from retrieved content. While we employ various techniques to reduce some of these risks, further work is needed to strengthen the system's robustness against such threats. We strongly recommend responsible deployment practices and ongoing monitoring when integrating our system into real-world applications.

Human Evaluation. Our human evaluation was carried out by anonymous volunteers who were fairly compensated for their time. All participants were fully informed about the purpose of the study and followed a structured annotation protocol. No personal data was collected or stored at any stage of the evaluation.

Chapter 10

Future Work

While PAKTON demonstrates strong performance across multiple dimensions of legal document analysis, it also opens up several promising avenues for future research and development. These directions aim to further enhance the system's accuracy, interpretability, robustness, and adaptability to a wider range of legal reasoning tasks and institutional deployment settings.

Ensemble Reasoning Across Language Models. One natural extension involves the integration of outputs from multiple language models—both open- and closed-source—through ensemble learning or model collaboration techniques. Instead of relying on a single LLM as the reasoning engine, future iterations of PAKTON could aggregate the outputs of several LLMs with diverse training backgrounds and reasoning styles. This approach may help mitigate individual model biases, compensate for knowledge gaps, and boost overall reliability. Techniques such as weighted voting, confidence-based aggregation, or meta-reasoning modules that select or synthesize answers from multiple candidates could be employed to reconcile outputs.

Multi-Agent Interrogation and Debate. Currently, the *Interrogator* in PAKTON is a single-agent component tasked with synthesizing information and formulating responses. A future enhancement would be to instantiate multiple Interrogators, each designed to represent a distinct interpretive stance or argumentative strategy. The system could implement mechanisms for structured debate, internal critique, or iterative refinement, with a final answer emerging from consensus-building, majority voting, or adversarial deliberation. This would not only enhance the depth and diversity of legal reasoning but also align with real-world legal discourse, which often hinges on competing interpretations.

Dynamic Knowledge Integration. As legal knowledge is constantly evolving, future versions of PAKTON could incorporate dynamic retrieval from up-to-date legal databases, case law, and regulatory corpora. This would involve integrating long-term memory mechanisms or temporal retrieval filters to ensure that responses reflect the most current legal standards and precedents. Additionally, tighter integration with structured knowledge graphs or case-based reasoning modules could improve fact-checking, citation accuracy, and relevance in complex legal scenarios.

Cross-Jurisdictional and Multilingual Adaptation. Another direction involves extending PAKTON's capabilities to handle multilingual and cross-jurisdictional legal content. This would require robust jurisdiction detection, legal domain adaptation, and fine-grained modeling of legal variation across different regulatory regimes. Leveraging translation pipelines, cross-lingual embeddings, and jurisdiction-aware prompt templates could help generalize PAKTON's architecture beyond its current English-centric implementation.

Evaluation Frameworks Aligned with Legal Judgment. Finally, although PAKTON has undergone rigorous human and automated evaluation, the broader community still lacks reliable, scalable metrics that reflect legal reasoning quality as assessed by domain experts. Developing new evaluation protocols—such as expert annotation pipelines, adjudication-based comparative frameworks, or simulation-based stress tests—could provide more nuanced and legally grounded assessments of future systems.

In summary, these directions point toward a broader vision of PAKTON not just as a tool for contract review, but as a flexible, evolving platform for high-assurance legal reasoning. By combining multi-agent deliberation, model ensembling, domain-specific retrieval, and interactive explanation, future iterations could push the boundaries of what legal AI systems can achieve in terms of fairness, reliability, and interpretability.

Chapter 11

Bibliography

[1] Achiam, O. J. et al. “GPT-4 Technical Report”. In: 2023. URL:

[2] Aires, J. P., Granada, R. L., and Meneguzzi, F. R. “ConCon: a contract conflict identifier”. In: *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems, 2019, Canadá.* 2019.

[3] Akarajaradwong, P., Pothavorn, P., Chaksangchaichot, C., Tasawong, P., Nopparatbundit, T., and Nutanong, S. “NitiBench: A Comprehensive Study of LLM Framework Capabilities for Thai Legal Question Answering”. In: *arXiv preprint arXiv:2502.10868* (2025).

[4] Anthropic. *Introducing Contextual Retrieval.* 2024.

[5] Anthropic. *Introducing the Model Context Protocol.* Accessed: 2025-04-17. Nov. 2024. URL:

[6] Anthropic. “Claude 3.7 Sonnet System Card”. In: 2025. URL:

[7] Argyrou, G., Dimitriou, A., Lymperaiou, M., Filandrianos, G., and Stamou, G. *Automatic Generation of Fashion Images using Prompting in Generative Machine Learning Models.* 2024. arXiv: `2407.14944 [cs.CV]`. URL:

[8] Ba, J. L., Kiros, J. R., and Hinton, G. E. *Layer Normalization.* 2016. arXiv: `1607.06450 [stat.ML]`.

[9] Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. “A Neural Probabilistic Language Model”. In: *J. Mach. Learn. Res.* 3.null (Mar. 2003), pp. 1137–1155. ISSN: 1532-4435.

[10] Bianchini, F., Calamo, M., De Luzi, F., Macrı, M., and Mecella, M. “Enhancing complex linguistic tasks resolution through fine-tuning LLMs, RAG and Knowledge Graphs (Short paper)”. In: *International Conference on Advanced Information Systems Engineering.* Springer. 2024, pp. 147–155.

[11] Brown, T. B. et al. *Language Models are Few-Shot Learners.* 2020. arXiv: `2005.14165 [cs.CL]`. URL:

[12] Brown, T. B. et al. *Language Models are Few-Shot Learners.* 2020. arXiv: `2005.14165 [cs.CL]`.

[13] Brown, T. B. et al. “Language Models are Few-Shot Learners”. In: *ArXiv* abs/2005.14165 (2020). URL:

[14] Chiang, C.-H. and Lee, H.-y. “Can Large Language Models Be an Alternative to Human Evaluations?” In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Ed. by A. Rogers, J. Boyd-Graber, and N. Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 15607–15631. DOI: `10.18653/v1/2023.acl-long.870`. URL:

[15] Colombo, P., Pires, T. P., Boudiaf, M., Melo, R., Culver, D., Morgado, S., Malaboeuf, E., Hautreux, G., Charpentier, J., and Desa, M. “SaulLM-54B & SaulLM-141B: Scaling Up Domain Adaptation for the Legal Domain”. In: *ArXiv* abs/2407.19584 (2024). URL:

[16] Colombo, P. et al. “SaulLM-7B: A pioneering Large Language Model for Law”. In: *ArXiv* abs/2403.03883 (2024). URL:

[17] Cormack, G. V., Clarke, C. L. A., and Büttcher, S. “Reciprocal rank fusion outperforms condorcet and individual rank learning methods”. In: *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (2009). URL:

[18] Cui, J., Li, Z., Yan, Y., Chen, B., and Yuan, L. “Chatlaw: A Multi-Agent Collaborative Legal Assistant with Knowledge Graph Enhanced Mixture-of-Experts Large Language Model”. In: 2023. URL:

[19] Deng, M., Wang, J., Hsieh, C.-P., Wang, Y., Guo, H., Shu, T., Song, M., Xing, E. P., and Hu, Z. *RLPrompt: Optimizing Discrete Text Prompts with Reinforcement Learning.* 2022. arXiv: `2205.12548 [cs.CL]`. URL:

[20] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *ArXiv* abs/1810.04805 (2019).

[21] Duarte, A. V., Marques, J. D., Graça, M., Freire, M., Li, L., and Oliveira, A. L. “LumberChunker: Long-Form Narrative Document Segmentation”. In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. Ed. by Y. Al-Onaizan, M. Bansal, and Y.-N. Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 6473–6486. DOI: `10.18653/v1/2024.findings-emnlp.377`. URL:

[22] Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., and Larson, J. “From Local to Global: A Graph RAG Approach to Query-Focused Summarization”. In: *ArXiv* abs/2404.16130 (2024). URL:

[23] Efrat, A. and Levy, O. *The Turking Test: Can Language Models Understand Instructions?* 2020. arXiv: `2010.11982 [cs.CL]`. URL:

[24] Eisenberg, M. A. “Legal Reasoning”. In: 2022. URL:

[25] Fan, W., Ding, Y., Ning, L.-b., Wang, S., Li, H., Yin, D., Chua, T.-S., and Li, Q. “A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models”. In: *Knowledge Discovery and Data Mining*. 2024. URL:

[26] Frei, T. “An Artificial Intelligence Approach To Legal Reasoning”. In: 2016. URL:

[27] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., and Wang, H. *Retrieval-Augmented Generation for Large Language Models: A Survey*. 2024. arXiv: `2312.10997 [cs.CL]`. URL:

[28] Gao, Y. et al. “Retrieval-Augmented Generation for Large Language Models: A Survey”. In: *ArXiv* abs/2312.10997 (2023). URL:

[29] Giadikiaroglou, P., Lymperaiou, M., Filandrianos, G., and Stamou, G. *Puzzle Solving using Reasoning of Large Language Models: A Survey*. 2024. arXiv: `2402.11291 [cs.CL]`. URL:

[30] Griogoriadou, N., Lymperaiou, M., Filandrianos, G., and Stamou, G. “AILS-NTUA at SemEval-2024 Task 6: Efficient model tuning for hallucination detection and analysis”. In: *ArXiv* abs/2404.01210 (2024). URL:

[31] Guo, Z., Xia, L., Yu, Y., Ao, T., and Huang, C. “LightRAG: Simple and Fast Retrieval-Augmented Generation”. In: *ArXiv* abs/2410.05779 (2024). URL:

[32] He, K., Zhang, X., Ren, S., and Sun, J. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: `10.1109/CVPR.2016.90`.

[33] Huang, Q., Tao, M., Zhang, C., An, Z., Jiang, C., Chen, Z., Wu, Z., and Feng, Y. “Lawyer llama technical report”. In: *arXiv preprint arXiv:2305.15062* (2023).

[34] Ichida, A. Y. and Meneguzzi, F. “Detecting Logical Relation In Contract Clauses”. In: *arXiv preprint arXiv:2111.01856* (2021).

[35] Institute for Supply Management. *Institute for Supply Management (ISM)*. Accessed: 2025-04-01.

[36] Kant, M., Nabi, S., Kant, M., Scharrer, R., Ma, M., and Nabi, M. “Towards robust legal reasoning: Harnessing logical llms in law”. In: *arXiv preprint arXiv:2502.17638* (2025).

[37] Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L. Y., Edunov, S., Chen, D., and Yih, W.-t. “Dense Passage Retrieval for Open-Domain Question Answering”. In: *ArXiv* abs/2004.04906 (2020). URL:

[38] Koreeda, Y. and Manning, C. “ContractNLI: A Dataset for Document-level Natural Language Inference for Contracts”. In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Ed. by M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1907–1919. DOI: `10.18653/v1/2021.findings-emnlp.164`. URL:

[39] Kritharoula, A., Lymperaiou, M., and Stamou, G. *Language Models as Knowledge Bases for Visual Word Sense Disambiguation*. 2023. arXiv: `2310.01960 [cs.CL]`. URL:

[40] Kritharoula, A., Lymperaiou, M., and Stamou, G. “Large Language Models and Multimodal Retrieval for Visual Word Sense Disambiguation”. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2023, pp. 13053–13077. DOI: `10.18653/v1/2023.emnlp-main.807`. URL:

[41] Lai, J., Gan, W., Wu, J., Qi, Z., and Yu, P. S. “Large language models in law: A survey”. In: *AI Open* (2024).

[42] LangChain. *Recursive Text Splitter*. Accessed: 2024-08-15. 2024.

[43] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.* 2019. arXiv: `1910.13461 [cs.CL]`.

[44] Lewis, P. et al. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *ArXiv* abs/2005.11401 (2020). URL:

[45] Li, Z., Peng, B., He, P., and Yan, X. "Evaluating the Instruction-Following Robustness of Large Language Models to Prompt Injection". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing.* Ed. by Y. Al-Onaizan, M. Bansal, and Y.-N. Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 557–568. DOI: `10.18653/v1/2024.emnlp-main.33`. URL:

[46] Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. "Lost in the Middle: How Language Models Use Long Contexts". In: *Transactions of the Association for Computational Linguistics* 12 (2023), pp. 157–173. URL:

[47] Liu, P., Liu, X., Yao, R., Liu, J., Meng, S., Wang, D., and Ma, J. "Hm-rag: Hierarchical multi-agent multimodal retrieval augmented generation". In: *arXiv preprint arXiv:2504.12330* (2025).

[48] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. *Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing.* 2021. arXiv: `2107.13586 [cs.CL]`. URL:

[49] Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., and Zhu, C. "G-Eval: NLG Evaluation using Gpt-4 with Better Human Alignment". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing.* Ed. by H. Bouamor, J. Pino, and K. Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 2511–2522. DOI: `10.18653/v1/2023.emnlp-main.153`. URL:

[50] Liu, Y., Jin, M., Pan, S., Zhou, C., Zheng, Y., Xia, F., and Yu, P. "Graph self-supervised learning: A survey". In: *IEEE Transactions on Knowledge and Data Engineering* (2022).

[51] Liusie, A., Manakul, P., and Gales, M. "LLM Comparative Assessment: Zero-shot NLG Evaluation through Pairwise Comparisons using Large Language Models". In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers).* Ed. by Y. Graham and M. Purver. St. Julian's, Malta: Association for Computational Linguistics, Mar. 2024, pp. 139–151. URL:

[52] Marques, N. L., Lamare Bastian-Pinto, C. de, and Brandão, L. E. T. "When contract clauses conflict: The Salvador light rail vehicle concession". In: *Case Studies on Transport Policy* 16 (2024), p. 101168.

[53] Mekala, R. S. R., Razeghi, Y., and Singh, S. "EchoPrompt: Instructing the Model to Rephrase Queries for Improved In-context Learning". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers).* Ed. by K. Duh, H. Gomez, and S. Bethard. Mexico City, Mexico: Association for Computational Linguistics, June 2024, pp. 399–432. DOI: `10.18653/v1/2024.naacl-short.35`. URL:

[54] Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. "Recurrent neural network based language model." In: *INTERSPEECH.* Ed. by T. Kobayashi, K. Hirose, and S. Nakamura. ISCA, 2010, pp. 1045–1048. URL:

[55] Nguyen, T., Bolton, L., Taesiri, M. R., and Nguyen, A. T. "HoT: Highlighted Chain of Thought for Referencing Supporting Facts from Inputs". In: *ArXiv* abs/2503.02003 (2025). URL:

[56] OpenAI. *Embedding Models.* Accessed: 2024-08-15. 2024.

[57] OpenAI. *Optimizing File Uploads in ChatGPT Enterprise.* Accessed: 2025-04-12. 2024. URL:

[58] Panagiotopoulos, I., Filandrianos, G., Lymperaiou, M., and Stamou, G. "RISCORE: Enhancing In-Context Riddle Solving in Language Models through Context-Reconstructed Example Augmentation". In: *Proceedings of the 31st International Conference on Computational Linguistics.* Ed. by O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, and S. Schockaert. Abu Dhabi, UAE: Association for Computational Linguistics, Jan. 2025, pp. 9431–9455. URL:

[59] Papadimitriou, C., Filandrianos, G., Lymperaiou, M., and Stamou, G. *Masked Generative Story Transformer with Character Guidance and Caption Augmentation.* 2024. arXiv: `2403.08502 [cs.CV]`. URL:

[60] Pipitone, N. and Alami, G. H. *LegalBench-RAG: A Benchmark for Retrieval-Augmented Generation in the Legal Domain.* 2024. arXiv: `2408.10343 [cs.AI]`. URL:

[61] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[62] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2020. arXiv: `1910.10683 [cs.LG]`.

[63] Raptopoulos, P., Filandrianos, G., Lymperaiou, M., and Stamou, G. "PAKTON: A Multi-Agent Framework for Question Answering in Long Legal Agreements". In: 2025. URL:

[64] Robertson, S. E. and Zaragoza, H. "The Probabilistic Relevance Framework: BM25 and Beyond". In: *Found. Trends Inf. Retr.* 3 (2009), pp. 333–389. URL:

[65] Ryu, C., Lee, S., Pang, S., Choi, C., Choi, H., Min, M., and Sohn, J.-Y. "Retrieval-based evaluation for LLMs: a case study in Korean legal QA". In: *Proceedings of the Natural Legal Language Processing Workshop 2023*. 2023, pp. 132–137.

[66] Saha, B., Saha, U., and Malik, M. Z. "Advancing Retrieval-Augmented Generation with Inverted Question Matching for Enhanced QA Performance". In: *IEEE Access* (2024).

[67] Shao, Y., Jiang, Y., Kanell, T. A., Xu, P., Khattab, O., and Lam, M. S. "Assisting in Writing Wikipedia-like Articles From Scratch with Large Language Models". In: *North American Chapter of the Association for Computational Linguistics*. 2024. URL:

[68] ShengbinYue, S., Huang, T., Jia, Z., Wang, S., Liu, S., Song, Y., Huang, X.-J., and Wei, Z. "Multi-Agent Simulator Drives Language Models for Legal Intensive Interaction". In: *Findings of the Association for Computational Linguistics: NAACL 2025*. 2025, pp. 6537–6570.

[69] Shu, D., Zhao, H., Liu, X., Demeter, D., Du, M., and Zhang, Y. "LawLLM: Law large language model for the US legal system". In: *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 2024, pp. 4882–4889.

[70] Siino, M., Falco, M., Croce, D., and Rosso, P. "Exploring LLMs Applications in Law: A Literature Review on Current Legal NLP Approaches". In: *IEEE Access* (2025).

[71] Smith, R. W. "An Overview of the Tesseract OCR Engine". In: *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)* 2 (2007), pp. 629–633. URL:

[72] Smith, R. W. "History of the Tesseract OCR engine: what worked and what didn't". In: *Electronic imaging*. 2013. URL:

[73] Sun, J., Dai, C., Luo, Z., Chang, Y., and Li, Y. "Lawluo: A chinese law firm co-run by llm agents". In: *arXiv preprint arXiv:2407.16252* (2024).

[74] Thomas, K., Filandrianos, G., Lymperaiou, M., Zerva, C., and Stamou, G. *"I Never Said That": A dataset, taxonomy and baselines on response clarity classification*. 2024. arXiv: `2409.13879 [cs.CL]`. URL:

[75] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL:

[76] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[77] Veturi, S., Vaichal, S., Jagadheesh, R. L., Tripto, N. I., and Yan, N. "Rag based question-answering for contextual response prediction system". In: *arXiv preprint arXiv:2409.03708* (2024).

[78] Wahidur, R. S., Kim, S., Choi, H., Bhatti, D. S., and Lee, H.-N. "Legal Query RAG". In: *IEEE Access* (2025).

[79] Wang, L., Chen, H., Yang, N., Huang, X., Dou, Z., and Wei, F. "Chain-of-Retrieval Augmented Generation". In: *ArXiv* abs/2501.14342 (2025). URL:

[80] Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., and Zhou, M. "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers". In: *Advances in neural information processing systems* 33 (2020), pp. 5776–5788.

[81] Wang, X. et al. "Searching for Best Practices in Retrieval-Augmented Generation". In: *Conference on Empirical Methods in Natural Language Processing*. 2024. URL:

[82] Wang, Y., Chen, L., Cai, S., Xu, Z., and Zhao, Y. "Revisiting Automated Evaluation for Long-form Table Question Answering". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Y. Al-Onaizan, M. Bansal, and Y.-N. Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 14696–14706. DOI: `10.18653/v1/2024.emnlp-main.815`. URL:

[83] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E. H., Xia, F., Le, Q., and Zhou, D. “Chain of Thought Prompting Elicits Reasoning in Large Language Models”. In: *ArXiv* abs/2201.11903 (2022). URL:

[84] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: 2201.11903 [cs.CL]. URL:

[85] Wen, Y., Jain, N., Kirchenbauer, J., Goldblum, M., Geiping, J., and Goldstein, T. *Hard Prompts Made Easy: Gradient-Based Discrete Optimization for Prompt Tuning and Discovery*. 2023. arXiv: 2302.03668 [cs.LG]. URL:

[86] Wikipedia contributors. *Feedforward neural network — Wikipedia, The Free Encyclopedia*. [Online; accessed 22-September-2022]. 2022.

[87] Wikipedia contributors. *Semi-supervised learning — Wikipedia, The Free Encyclopedia*. [Online; accessed 22-September-2022]. 2022.

[88] Wiratunga, N., Abeyratne, R., Lasal Jayawardena, K. M., Massie, S., Nkisi-Orji, I., Weerasinghe, R., Liret, A., and Fleisch, B. *CBR-RAG: Case-Based Reasoning for Retrieval Augmented Generation in LLMs for Legal Question Answering.” arXiv, April 4, 2024.*

[89] World Commerce & Contracting. *The Real Value of Contracting*. Accessed: 2025-04-01. 2020.

[90] Xu, X., Tao, C., Shen, T., Xu, C., Xu, H., Long, G., Lou, J.-G., and Ma, S. “Re-Reading Improves Reasoning in Large Language Models”. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Y. Al-Onaizan, M. Bansal, and Y.-N. Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 15549–15575. DOI: 10.18653/v1/2024.emnlp-main.871. URL:

[91] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. “ReAct: Synergizing Reasoning and Acting in Language Models”. In: *ArXiv* abs/2210.03629 (2022). URL:

[92] Zhang, K., Yu, W., Sun, Z., and Xu, J. “SyLeR: A Framework for Explicit Syllogistic Legal Reasoning in Large Language Models”. In: *arXiv preprint arXiv:2504.04042* (2025).

[93] Zhao, J., Zu, C., Hao, X., Lu, Y., He, W., Ding, Y., Gui, T., Zhang, Q., and Huang, X.-J. “LONGAGENT: Achieving Question Answering for 128k-Token-Long Documents through Multi-Agent Collaboration”. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 2024, pp. 16310–16324.

[94] Zhao, X., Zhou, X., and Li, G. “Chat2data: An interactive data analysis system with rag, vector databases and llms”. In: *Proceedings of the VLDB Endowment* 17.12 (2024), pp. 4481–4484.

[95] Zhou, H., Zhou, L., Gao, B., Huang, W., Huang, W., Zuo, J., and Zhao, X. “Intelligent identification of risks in construction contract clauses based on semantic reasoning”. In: *Engineering, Construction and Architectural Management* (2024).

This chapter provides supplementary material and additional details that support and extend the discussions presented in the preceding chapters.

Appendix A

Framework's Implementation Details

A.1 Prompts

Throughout the framework, various prompts are employed across multiple stages to guide the system's behavior. Due to space constraints, it is not feasible to include all of them in this appendix. Instead, we provide the prompts used in one of the core components of the framework—the generation of interrogation questions.

INTERROGATION SYSTEM PROMPT

You are a skilled legal interrogator conducting an in-depth interview with a legal researcher. Your objective is to extract **comprehensive, well-supported legal information** by formulating precise, strategic questions.
The goal is **not simply to obtain answers**, but to gather authoritative legal evidence, reasoning, and precedents to thoroughly address the following legal question:
<question>{userQuery} </question>
Additional Context: The following background information relevant to the question is provided:
<context>{userContext} </context>
Additional Instructions: You must take into account the following instructions:
<intructions>{userInstructions} </intructions>
—
Critically Consider the Existing Report Before Asking New Questions:
You have been provided with a **report summarizing the interrogation so far**. This report serves as a **synthesis of key legal arguments, findings, acknowledged knowledge gaps, and preliminary reasoning** extracted from the conversation.
Before forming your next question, **carefully analyze this report**, which includes:
- The **preliminary reasoning and draft interpretation**—a tentative legal direction that has emerged, but is still subject to revision.
- **Explicitly acknowledged knowledge gaps**—areas where the legal researcher did not provide sufficient clarity, evidence, or citations.
- **Remaining uncertainties and conflicting viewpoints**, including legal areas where additional investigation is required.
- **Follow-up questions that have already been identified** to refine the legal analysis further.

You must use this information **strategically** to craft your next question.
—
Your Role:
- You have **{remaining_questions} questions remaining**, so each question must be maximally informative.
- Your goal is to **clarify uncertainties, challenge assumptions, and press for concrete legal sources** to fill the knowledge gaps.
- Your questions should probe deeper into weak or vague responses, pressing for **specific legal precedents, case law, statutory references, and counterarguments**.
- Avoid redundancy—do not ask questions that have already been answered in the report. Instead, **build upon previous insights** and push the conversation forward.

—
...
—
Completion: Once you are fully satisfied that you have gathered all necessary legal insights, you may conclude the interrogation by stating: *"Thank you, I am now in a position to answer the question with confidence."*
You will be given:
- The report summarizing the previous exchange with the legal researcher.
- The list of previous questions asked so far.

Use this information to ensure your next question is targeted, strategic, and maximally informative.

INTERROGATION_USER_PROMPT

The following report summarizes the previous exchange between you and the legal researcher.
<report>{report} </report>
This report contains:
- **A preliminary interpretation or draft answer**, which is subject to revision.
- **Explicitly acknowledged gaps in legal reasoning**—areas that require further clarification.
- **Conflicting viewpoints or legal uncertainties** that need to be resolved.
- **Follow-up questions that have been identified** to improve the legal analysis.

The following questions have been asked so far:
<questions>{questions} </questions>

You must carefully analyze the above report before crafting your next question.
Your next question should:
- **Push the conversation forward**—do not repeat questions that have already been asked.
- **Target unresolved knowledge gaps** and press for **specific legal references**.
- **Challenge weak or unsupported reasoning**—seek case law, statutes, or counterarguments.
- **Refine or reassess the preliminary interpretation**, if needed.
- **Help move toward a stronger, well-supported legal answer**.

Now, continue your interrogation.

Appendix B

Experiments on LegalBenchRAG

B.1 Baselines

In the original LegalBenchRAG paper, the authors evaluated several retrieval configurations for in-document retrieval using the LegalBench benchmark. The configurations and their corresponding experimental setups are summarized below:

- **Method 1**: Naive fixed-size chunking with a window of 500 characters and no overlap; no reranker; embeddings generated using `text-embedding-3-large`. Results reported in Table 7.2 as "Naive".
- **Method 2**: Recursive Character Text Splitter (RCTS) with no overlap; no reranker; embeddings generated using `text-embedding-3-large`. Results reported in Table 7.2 as "RCTS".
- **Method 3**: Naive fixed-size chunking (500 characters, no overlap) with reranking using the Cohere reranker `rerank-english-v3.0`; embeddings from `text-embedding-3-large`. Results reported in Table 7.2 as "Naive + Cohere".
- **Method 4**: Recursive Character Text Splitter with no overlap and reranking using `rerank-english-v3.0`; embeddings from `text-embedding-3-large`. Results reported in Table 7.2 as "RCTS + Cohere".

Among these, Method 2 was selected by the authors as the preferred configuration. However, in this work we compare PAKTON against all four baseline configurations to demonstrate consistent performance improvements.

B.2 PAKTON Evaluation Strategy

To assess the retrieval capabilities of the **PAKTON** system on LegalBenchRAG, we evaluated the indexing and in-document retrieval functionality, which constitute the core components of the pipeline. In this setup:

- The **Archivist** component is responsible for document indexing.
- The **Researcher** performs in-document retrieval.
- Interactions through the **Interrogator** were bypassed to focus exclusively on retrieval performance.

Each document from the dataset was indexed by the Archivist, and the corresponding queries were directly submitted to the Researcher. The retrieved spans were evaluated using the LegalBenchRAG scoring methodology.

Configuration 1

Archivist:

- Primary strategy: structural parsing.
- Fallback: Recursive Character Text Splitter with 1000-character chunks and no overlap if structural parsing failed.
- Embedding model: `text-embedding-3-large`.

Researcher:

- Query optimization using gpt-4o and only in-document search as tool enabled.
- BM25 retrieves top-100 chunks with a similarity threshold of 0.6.
- Dense embedding retriever returns top-100 chunks with no similarity filtering using embeddings model `text-embedding-3-large`
- Reciprocal Rank Fusion with equal weights for both retrievers to rerank chunks, pick top-64 chunks.
- Reranker: `BAAI/bge-reranker-v2-m3`, producing a top-64 reranked final list of chunks.
- Strip structural information of the chunk and keep only the original span.

LLM Filtering:

- An additional post-reranking filtering stage is applied using `command-R` (Cohere), an open-source 35B parameter model specifically fine-tuned for Retrieval-Augmented Generation (RAG) applications.
- From the top-10 reranked chunks, the model identifies and extracts the most relevant sub-span(s) within each chunk, aiming to isolate highly precise evidence.
- This step is designed to enhance overall precision by focusing retrieval results on the most contextually pertinent portions of the content.

Results for this configuration are provided in Table B.1. Two variants are compared: one with and one without the LLM filtering step. The results reported and compared in Table 7.2 correspond to Configuration 1, specifically the variant without LLM filtering.

Configuration 2

Same as Configuration 1 with the only change being the use of an alternative LLM-based reranker:

- Reranker: `AAI/bge-reranker-v2-minicpm-layerwise` (2.72B parameters) with a 28-layer cutoff.

Results for this configuration are provided in Table B.2.

B.3 Discussion

Configuration 1 vs Configuration 2: While Configuration 1 serves as the primary setup for PAKTON, our evaluation indicates that Configuration 2 yields superior performance in terms of retrieval accuracy. However, this improvement comes at a cost: the second reranker in Configuration 2 is a large language model with 2.72 billion parameters, which introduces a significant latency overhead compared to the more lightweight setup in Configuration 1.

Considering the trade-off between reranking accuracy and computational efficiency, Configuration 1 represents the most suitable choice for practical deployment scenarios where speed is a critical factor. Nevertheless, for applications where performance is prioritized over inference time, Configuration 2 may be preferred to achieve more competitive results.

No LLM Filtering vs. LLM Filtering: In the legal domain, **high recall** is often of paramount importance, as omitting relevant spans can lead to incomplete or flawed legal reasoning. The *No LLM Filtering* configuration aligns better with this paradigm minimizing the risk of excluding potentially critical information. For this reason, it is selected as the primary setup in our evaluation.

Nonetheless, there are scenarios where **precision** is more desirable—particularly when mitigating hallucination risks or when users require concise, targeted evidence rather than exhaustive retrieval. To accommodate such use cases, we evaluate an additional post-reranking stage employing the *LLM Filtering* variant, which identifies and extracts the most relevant sub-spans within each of the top-10 reranked chunks, thereby filtering the retrieval output to focus on the most contextually pertinent segments.

As shown in Tables B.1 and B.2, this LLM Filtering variant significantly improves **precision**. However, this gain comes at the cost of **recall**, especially as the value of `top_k` increases, due to the stricter content selection. Based on these findings, we conclude that LLM Filtering is particularly advantageous for low `top_k` settings, where focused and precise evidence is preferred. In contrast, for higher `top_k` values, the unfiltered setup is more appropriate to maintain broader recall.

Character-Based vs. Span-Based Calculation of Precision and Recall: The LegalBench-RAG paper adopts a character-based approach for computing precision and recall, and we follow the same protocol for our primary evaluation. However, we observe that this method may penalize retrieval strategies—particularly those targeting precision like the *LLM Filtering* variant—due to the fine-grained nature of the retrieved spans. Specifically, in many cases the retrieved content consists of subspans (often smaller than a sentence) that lie within the annotated answer span. Under the character-based metric, such partial matches are treated as incomplete, thereby reducing recall—even when the retrieved content is semantically relevant and informative.

To further investigate this effect, we introduce a complementary *span-based* evaluation. In this setting, a retrieved span is considered a *hit* if it overlaps with any ground truth span, and a *miss* otherwise. This binary overlap-based metric provides clearer insight into how often irrelevant spans are retrieved or relevant spans are entirely missed. The corresponding results are reported in Table B.3 and should be interpreted in comparison with the character-based results in Table B.1.

Overall, the span-based evaluation yields consistently higher values, with a particularly notable improvement in recall for the *LLM Filtering* variant. This suggests that character-level metrics may disproportionately penalize methods optimized for precision, potentially underestimating their effectiveness. It is also important to consider that different datasets contain varying numbers of ground truth spans per example, which can limit achievable recall at low `top_k` values. For reference, Table B.4 presents the upper bounds on recall achievable under 100% precision.

Variation on Character Volume per chunk for different retrieval methods: Our analysis of precision and recall is conducted across varying `top_k` values. However, it is important to note that different retrieval methods return varying volumes of text, even when the same number of chunks is retrieved. For instance, one method may retrieve significantly fewer characters on average per chunk compared to another, despite retrieving the same number of chunks. To better understand this disparity, Table B.5 reports the total number of characters retrieved at various `k` values, the average answer length, and the average document length for each dataset.

This analysis allows us to assess the actual amount of information passed to the LLM across methods. Notably, the *LLM Filtering* variant consistently retrieves fewer characters than its unfiltered counterpart for the same number of chunks.

Moreover, this analysis provides insight into the degree of document compression achieved during retrieval. For example, in the MAUD dataset, our *span-based* Recall@64 reaches 85.45% while retrieving, on average, 56,523 characters—compared to an average document length of 353,718 characters—indicating an approximate 84% compression of the original document content. Similarly, in the ContractNLI dataset, the *LLM Filtering* variant achieves a Recall@32 of 74.41% using only 1,081 characters, which corresponds to nearly 90% information compression.

Plots. Figure B.3.1 presents the evaluation plots for all four methods of LegalBenchRAG and PAKTON, including Precision@k, Recall@k, and Precision–Recall curves across all datasets. The results indicate that PAKTON consistently outperforms the other methods across all metrics.

Conclusion: Taking into account the span-based evaluation metrics—which provide a more representative measure of retrieval quality—and the observed degree of information compression, the *Researcher* module demonstrates strong performance in the task of long-document retrieval within a practical deployment context.

		Precision @ k							Recall @ k						
Dataset	**LLM filtering**	1	2	4	8	16	32	64	1	2	4	8	16	32	64
PrivacyQA	without	19.94	16.84	11.44	8.62	7.38	6.42	6.08	13.34	22.43	32.67	43.39	61.65	82.30	89.42
	with	29.11	30.33	25.99	23.35	22.64	22.33	22.33	10.51	18.13	23.33	26.35	27.65	28.19	28.19
ContractNLI	without	33.02	30.34	17.33	9.98	5.87	4.68	4.52	53.14	67.47	80.06	89.71	95.50	99.56	99.82
	with	59.59	51.36	46.32	45.00	45.00	44.87	44.87	38.53	45.25	51.95	54.94	58.00	58.69	58.69
MAUD	without	25.47	17.45	10.51	7.24	5.08	3.18	1.85	23.99	30.09	34.49	46.42	59.74	74.96	82.80
	with	38.87	36.99	33.54	33.12	32.77	32.33	32.29	19.06	22.60	24.06	26.52	27.51	27.64	27.64
CUAD	without	11.02	8.83	6.81	4.72	2.78	2.07	1.62	16.52	24.76	33.34	46.67	59.53	77.08	86.23
	with	29.14	29.53	29.31	28.86	28.89	28.79	28.77	25.31	30.30	34.33	37.68	38.24	38.68	38.68
ALL	without	**22.34**	**18.37**	**11.52**	**7.63**	**5.26**	**4.08**	**3.52**	**26.77**	**36.32**	**45.26**	**56.66**	**69.17**	**83.50**	**89.58**
	with	**39.17**	**37.03**	**33.78**	**32.58**	**32.26**	**32.08**	**32.05**	**23.37**	**29.07**	**33.42**	**36.36**	**37.84**	**38.29**	**38.29**

Table B.1: Performance comparison on different datasets for Precision and Recall at various k values for PAKTON's Researcher and Archivist under configuration 1.

		Precision @ k							Recall @ k						
Dataset	**LLM filtering**	1	2	4	8	16	32	64	1	2	4	8	16	32	64
PrivacyQA	without	18.64	17.66	14.13	10.39	7.51	6.34	6.01	17.00	26.39	39.99	62.73	79.24	94.90	97.35
	with	33.02	31.85	29.33	26.66	26.11	25.70	25.68	10.35	18.20	23.12	26.95	28.20	28.58	28.76
ContractNLI	without	**42.21**	**36.86**	**19.81**	**10.58**	**6.13**	**4.63**	**4.53**	**67.67**	**83.35**	**90.87**	**95.16**	**98.19**	**99.81**	**100.00**
	with	77.51	67.15	64.02	63.02	63.22	63.23	63.23	51.78	58.93	63.10	64.52	65.07	65.24	65.24
MAUD	without	17.77	11.35	9.41	8.32	5.84	3.50	1.92	17.06	19.94	30.57	49.94	68.82	80.10	86.51
	with	37.71	33.60	31.01	32.12	32.11	31.96	31.92	17.92	21.62	24.69	28.02	29.90	30.48	30.49
CUAD	without	2.61	5.10	5.72	4.18	2.84	1.84	1.29	4.38	16.11	34.09	53.00	69.13	80.65	86.20
	with	25.51	28.62	30.28	31.13	30.27	30.21	30.22	15.54	23.74	30.06	33.11	33.60	33.60	33.65
ALL	without	20.31	17.74	12.27	8.37	5.58	4.08	3.44	26.53	36.45	48.88	65.21	78.845	88.87	92.52
	with	43.44	40.30	38.66	38.23	37.93	37.78	37.76	23.90	30.62	35.24	38.15	39.19	39.48	39.54

Table B.2: Performance comparison across different datasets in terms of Precision and Recall at various k values, using PAKTON's Researcher and Archivist under Configuration 2.

		Precision @ k							Recall @ k						
Dataset	**LLM filtering**	1	2	4	8	16	32	64	1	2	4	8	16	32	64
PrivacyQA	without	35.08	30.37	23.69	17.86	14.79	11.73	10.88	19.65	32.23	43.52	58.46	75.93	89.34	94.18
	with	33.51	32.98	28.01	26.16	25.61	25.43	25.39	20.36	31.65	37.62	42.49	43.94	46.89	46.89
ContractNLI	without	58.76	39.69	24.10	13.63	7.86	5.99	5.87	53.74	68.56	80.50	89.86	95.45	99.48	99.74
	with	63.73	54.12	49.40	48.01	47.90	47.81	47.81	54.10	61.30	68.17	71.60	72.90	73.41	73.41
MAUD	without	35.05	23.97	15.34	11.08	8.02	5.27	3.08	32.32	38.24	43.56	54.98	67.17	81.12	85.45
	with	41.58	36.01	32.22	31.34	30.72	30.33	30.29	36.90	41.18	44.18	47.77	48.55	48.81	48.81
CUAD	without	20.10	15.46	10.82	8.13	5.30	3.91	3.08	16.86	24.98	33.95	47.84	60.85	77.66	84.38
	with	35.75	30.93	29.38	29.11	28.83	28.82	28.81	30.99	37.81	43.14	47.04	48.85	49.28	49.28
ALL	without	37.24	27.37	18.49	12.68	8.99	6.73	5.73	30.64	41.00	50.38	62.79	74.85	86.90	90.94
	with	43.64	38.51	34.75	33.66	33.27	33.10	33.08	35.59	42.99	48.28	52.23	53.56	54.60	54.60

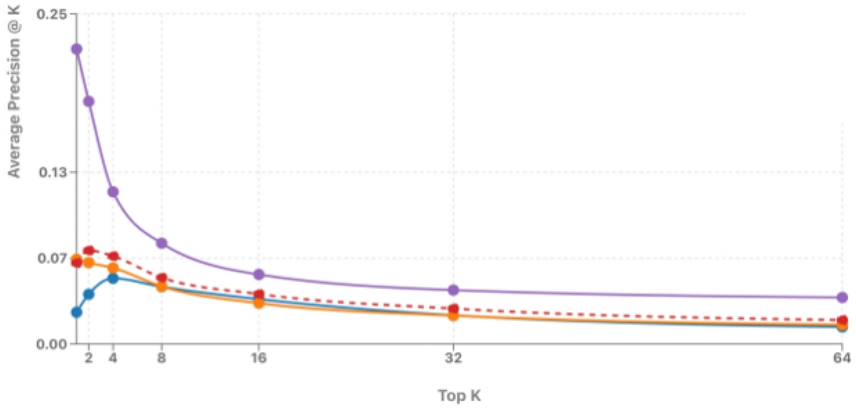
Table B.3: Performance comparison across different datasets in terms of Precision and Recall at various k values, using PAKTON's Researcher and Archivist components under Configuration 1, based on span-based calculation of Precision and Recall.

	Recall @ k						
Dataset	1	2	4	8	16	32	64
PrivacyQA	61.91	86.97	96.42	99.12	100.00	100.00	100.00
ContractNLI	82.16	98.69	100.00	100.00	100.00	100.00	100.00
MAUD	68.19	92.35	98.21	99.86	100.00	100.00	100.00
CUAD	82.14	94.30	98.44	100.00	100.00	100.00	100.00
ALL	**73.63**	**93.09**	**98.27**	**99.75**	**100.00**	**100.00**	**100.00**

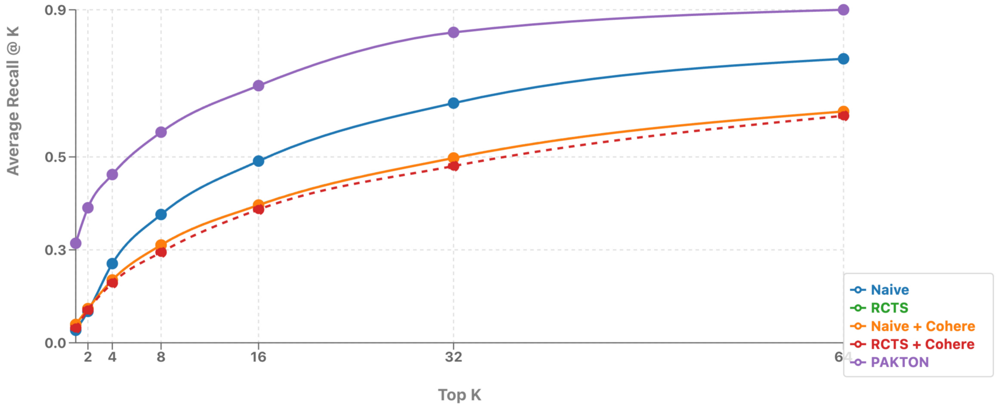
Table B.4: Recall @ k for perfect retrieval using ground-truth snippets. Note: Precision is 100% for all cases.

		Number of Characters @ k							Avg. Lengths	
Dataset	**LLM Filtering**	1	2	4	8	16	32	64	**Answer**	**Document**
PrivacyQA	without	682	1415	2985	5706	10341	17961	21080		
	with	207	403	711	1088	1362	1476	1523	1200	25266
ContractNLI	without	789	1371	2738	4869	8126	10774	11112		
	with	250	412	651	885	1036	1081	1081	438	10782
MAUD	without	908	1765	3542	7069	14248	28699	56523		
	with	301	477	678	906	1027	1059	1071	1259	353718
CUAD	without	771	1643	3496	7002	13694	24057	35750		
	with	250	383	527	749	844	896	971	600	69303

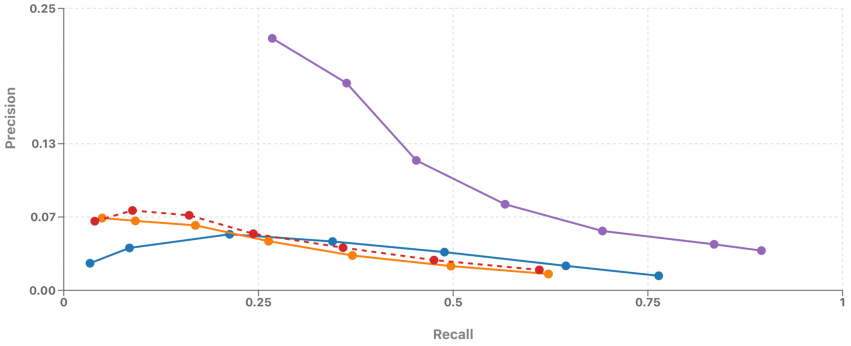
Table B.5: Average number of characters retrieved @ k for each dataset for PAKTON configuration 1. Comparison with average length of ground truth (answer) and document lengths.



(a) Precision @ k across all datasets



(b) Recall @ k across all datasets



(c) Precision - Recall across all datasets

Figure B.3.1: Precision and Recall values for different k across all datasets for all methods

Appendix C

Experiments on ContractNLI

C.1 Dataset Overview

We conduct all experiments on the test split of the `ContractNLI` dataset[38], which contains 2,091 samples. Each sample consists of a full non-disclosure agreement (NDA) as the *premise*, a legal statement as the *hypothesis*, and an *inference label* indicating whether the hypothesis is entailed, contradicted, or neutral with respect to the contract.

Data Access. We use the Hugging Face implementation of the dataset: https://huggingface.co/datasets/kiddothe2b/contract-nli

Predictions. For each experimental configuration, predictions and their associated data points are stored in structured `.json` files, which will be made publicly available upon publication.

C.2 Dataset Subsets

- `contractnli_b:` The full version of the dataset. It includes 7.19K samples in the training split, 1.04K in the validation split, and 2.09K in the test split. The full contract is used as the premise for each example.
- `contractnli_a:` A filtered version of `contractnli_b` in which only the minimal spans necessary to determine the correct label are retained as the premise, significantly reducing the input length. Experiments using this subset simulate ideal retrieval conditions and serve as an upper bound for the potential performance of a perfect RAG system.

C.3 Prompting Strategies

We experiment with the following prompting techniques:

- `naive zero-shot (ZS):` No examples are given. A basic description of the label classes is provided, and the full contract is used as the premise. This serves as a solid baseline for performance.
- `optimized zero-shot (opt. ZS):` Uses hardcoded explanations of the classes and improved prompt structure. Still uses the full contract as the premise. Explores the effect of manual prompt engineering.
- `naive few-shot (FS):` Based on [13], builds on the optimized zero-shot format, but includes three random training examples (contract, hypothesis, and label) in the prompt. The full contract is used as the premise in both the examples and the current input. Demonstrates the effect of using cross-document retrieval.

- `naive few-shot isolated spans (FS+Spans):` Similar to naive few-shot, but in the training examples, only the relevant spans (rather than the entire contract) are provided as the premise.
- `naive few-shot isolated spans (same hypothesis) (FS+Spans+Hyp):` A refinement of the above method, in which few-shot examples are dynamically selected to match the hypothesis of the current sample, increasing semantic alignment. Shows the effect of using effective cross-document retrieval and chunking setting the upper boundary for the ideal cross-document RAG system.
- `Chain of Thought (CoT):` Based on [83], extends the previous method by also including reasoning steps (i.e., rationales or justifications) in the answers of the few-shot examples, encouraging more explicit reasoning in the final output. Examines the impact of explicitly articulating the reasoning behind decisions.

C.4 Experimental Setup

Hardware: All local experiments were conducted on a server equipped with 4x NVIDIA A6000 GPUs, each with 48GB VRAM, using the AI daptiv infrastructure.

Quantization: We employed multiple quantization levels to evaluate performance under varying resource constraints. The specific techniques and configurations used will be made available in our GitHub repository upon publication.

Scale: In total, we ran 210 distinct experiments (each one for the whole test set of ContractNLI) across model, prompting, and input configurations.

C.5 Detailed Results

All the experiments and the corresponding results are shown in detail in Tables C.1, C.2, C.3, C.4, C.5 and C.6.

C.6 Execution Details of PAKTON

To simulate a *cross-document retrieval* setting, we indexed the training and validation splits of the ContractNLI dataset, preserving each example alongside its corresponding ground-truth label. Each contract chunk was embedded and stored in the most appropriate index, where grouping was determined by the combination of the hypothesis and its associated label. This approach emulates the core behavior of the *Archivist* module, which supports organizing textual segments into logically distinct indices—such as by contract type or clause category.

For example, all instances associated with the hypothesis “The Receiving Party shall not disclose the fact that the Agreement was agreed or negotiated” and labeled as *Neutral* were stored within a single index, while examples labeled as *Entailment* or *Contradiction* were assigned to their respective indices.

Each set of indices corresponding to the same hypothesis was interconnected into a composable graph using the `LlamaIndex` framework. These hypothesis-specific graphs were then integrated into a unified, higher-level composable graph. Every node—whether a graph or a leaf index—was annotated with a brief natural language description summarizing the content it encapsulated.

At inference time, this hierarchical structure was traversed recursively. At each level of the graph, a similarity comparison was conducted between the input query and the textual descriptions of child nodes to determine the most relevant subgraph to explore. This hierarchical traversal mechanism enables efficient prioritization of semantically aligned indices, thereby improving retrieval relevance.

The *Researcher* module utilized this architecture as a cross-document retrieval system, returning the top-3 most relevant examples for a given query.

As for the *Researcher* module, we utilized **Configuration 1** (see Section B.2), with *No LLM filtering* and kept the top-10 reranked chunks to generate the response back to the Interrogator.

Regarding the interrogation process, we capped the maximum number of turns to five in order to maintain efficiency and avoid excessively long interaction sequences.

No	**Model ID**	**Acc.**	**F1[W]**	**F1[E]**	**F1[C]**	**F1[N]**	**Quant.**	**Try**	**Prompting**
0	Llama-3-8B-Instr	0.4868	0.4612	0.5677	0.0327	0.4514	no	1	ZS
1	Llama-3-8B-Instr	0.4883	0.4613	0.5639	0.0252	0.4576	no	2	ZS
2	Llama-3-8B-Instr	0.4969	0.4691	0.5758	0.0166	0.4651	no	3	ZS
3	Mistral-7B-Instr	0.5301	0.4804	0.4551	0.0261	0.6178	no	1	ZS
4	Mistral-7B-Instr	0.5340	0.4843	0.4627	0.0175	0.6206	no	2	ZS
5	Mistral-7B-Instr	0.5328	0.4839	0.4590	0.0342	0.6201	no	3	ZS
6	Saul-7B-Instr	0.4223	0.2933	0.0570	0.0910	0.5960	no	1	ZS
7	Saul-7B-Instr	0.4218	0.2960	0.0760	0.0542	0.5907	no	2	ZS
8	Saul-7B-Instr	0.4146	0.2809	0.0437	0.0588	0.5893	no	3	ZS
9	Mixtral-8x7B-Instr	0.5608	0.5659	0.6548	0.4087	0.5088	normal	1	ZS
10	Mixtral-8x7B-Instr	0.5605	0.5652	0.6551	0.4099	0.5065	normal	2	ZS
11	Mixtral-8x7B-Instr	0.5612	0.5661	0.6545	0.4105	0.5092	normal	3	ZS
12	Llama-3-8B-Instr	0.5151	0.4670	0.6331	0.0000	0.4026	no	1	opt. ZS
13	Llama-3-8B-Instr	0.5227	0.4738	0.6372	0.0000	0.4140	no	2	opt. ZS
14	Llama-3-8B-Instr	0.5261	0.4785	0.6384	0.0000	0.4237	no	3	opt. ZS
15	Llama-3-8B-Instr	0.5332	0.5013	0.6144	0.0090	0.5000	no	1	FS+Spans
16	Llama-3-8B-Instr	0.5232	0.4938	0.5977	0.0177	0.4986	no	2	FS+Spans
17	Llama-3-8B-Instr	0.5333	0.5024	0.6089	0.0089	0.5086	no	3	FS+Spans
18	SaulLM-54B-Instr	0.7021	0.6806	0.7726	0.1832	0.7032	lowest	1	ZS
19	SaulLM-54B-Instr	0.7001	0.6764	0.7724	0.1594	0.6995	lowest	2	ZS
20	SaulLM-54B-Instr	0.7040	0.6807	0.7731	0.1760	0.7046	lowest	3	ZS
21	Llama-3-70B-Instr	0.6241	0.5990	0.7313	0.4972	0.4819	normal	1	ZS
22	Llama-3-70B-Instr	0.6236	0.5987	0.7320	0.4943	0.4813	normal	2	ZS
23	Llama-3-70B-Instr	0.6208	0.5953	0.7297	0.4943	0.4759	normal	3	ZS
24	DS-R1-Dist.-Llama-70B	0.6054	0.6078	0.6552	0.4837	0.5874	normal	1	ZS
25	gpt-4o	0.6112	0.6367	0.7497	0.4119	0.5704	-	1	ZS
26	gpt-4o	0.6112	0.6356	0.75	0.4151	0.5667	-	2	ZS
27	gpt-4o	0.6141	0.6376	0.7474	0.4216	0.5725	-	3	ZS
28	gpt-4o	0.6275	0.6265	0.5957	0.4416	0.7046	-	1	opt. ZS
29	gpt-4o	0.6165	0.6169	0.5916	0.4455	0.6856	-	2	opt. ZS
30	gpt-4o	0.6227	0.6240	0.5973	0.4364	0.6983	-	3	opt. ZS
31	gpt-4o	0.6523	0.6693	0.7315	0.4577	0.6542	-	1	FS
32	gpt-4o	0.6590	0.6733	0.7297	0.4624	0.6642	-	2	FS
33	gpt-4o	0.6805	0.6941	0.7503	0.5	0.6813	-	3	FS
34	gpt-4o	0.6413	0.6507	0.6556	0.4533	0.6936	-	1	FS+Spans
35	gpt-4o	0.6538	0.6627	0.6724	0.4719	0.6989	-	2	FS+Spans
36	gpt-4o	0.6495	0.6587	0.6712	0.4656	0.6924	-	3	FS+Spans
37	gpt-4o	0.6906	0.7108	0.7919	0.4866	0.6784	-	1	ZS RAG
38	Llama3-8b-Instr	0.4776	0.4278	0.6029	0.0536	0.3314	-	1	ZS
39	Mistral-7b-Instr	0.527	0.4876	0.4863	0.0248	0.6018	-	1	ZS
40	Mistral-8x7b-Instr	0.5872	0.5948	0.656	0.4427	0.5664	-	1	ZS

Table C.1: Baseline performance of models across multiple evaluation runs on the ContractNLI test set. (Part 1 of 3)

No	**Model ID**	**Acc.**	**F1[W]**	**F1[E]**	**F1[C]**	**F1[N]**	**Quant.**	**Try**	**Prompting**
41	Llama3-70b-Instr	0.571	0.536	0.4493	0.3969	0.6628	-	1	ZS
42	Claude-3-Opus	0.7547	0.7676	0.8339	0.5083	0.7596	-	1	ZS
43	Claude-3-Opus	0.7461	0.7592	0.8279	0.4992	0.7489	-	2	ZS
44	Claude-3-Opus	0.7475	0.7608	0.8319	0.4970	0.7489	-	3	ZS
45	Claude-3.5-Sonnet	0.7944	0.8002	0.8757	0.5789	0.7732	-	1	ZS
46	Claude-3.5-Sonnet	0.7881	0.7949	0.8749	0.5609	0.7661	-	2	ZS
47	Claude-3.5-Sonnet	0.7924	0.7981	0.8764	0.5770	0.7679	-	3	ZS
48	Claude-3-Opus	0.7819	0.7891	0.8432	0.5622	0.7863	-	1	opt. ZS
49	Claude-3-Opus	0.7857	0.7935	0.8502	0.5559	0.7906	-	2	opt. ZS
50	Claude-3-Opus	0.7819	0.7886	0.8441	0.5606	0.7847	-	3	opt. ZS
51	Claude-3.5-Sonnet	0.7901	0.7960	0.8676	0.5710	0.7740	-	1	opt. ZS
52	Claude-3.5-Sonnet	0.7905	0.7967	0.8643	0.5728	0.7787	-	2	opt. ZS
53	Claude-3.5-Sonnet	0.7915	0.7975	0.8694	0.5681	0.7762	-	3	opt. ZS
54	Claude-3-Opus	0.7580	0.7593	0.8382	0.5628	0.7226	-	1	FS
55	Claude-3-Opus	0.7676	0.7680	0.8410	0.5945	0.732	-	2	FS
56	Claude-3-Opus	0.7542	0.7588	0.8308	0.5406	0.7348	-	3	FS
57	Claude-3.5-Sonnet	0.7709	0.7752	0.8542	0.5654	0.7415	-	1	FS
58	Claude-3.5-Sonnet	0.7838	0.7873	0.8645	0.5781	0.7556	-	2	FS
59	Claude-3.5-Sonnet	0.7786	0.7824	0.8576	0.5671	0.7543	-	3	FS
60	Claude-3-Opus	0.7862	0.7919	0.8489	0.5673	0.7855	-	1	FS+Spans
61	Claude-3-Opus	0.7891	0.7944	0.8457	0.5964	0.7876	-	2	FS+Spans
62	Claude-3-Opus	0.7786	0.7856	0.8393	0.5575	0.7836	-	3	FS+Spans
63	Claude-3.5-Sonnet	0.7977	0.8012	0.8650	0.6032	0.7811	-	1	FS+Spans
64	Claude-3.5-Sonnet	0.8015	0.8048	0.8717	0.6049	0.7819	-	2	FS+Spans
65	Claude-3.5-Sonnet	0.8006	0.8040	0.8668	0.6056	0.7850	-	3	FS+Spans
66	Claude-3-Opus	0.7752	0.7762	0.8437	0.5900	0.7492	-	1	FS+Spans+Hyp
67	Claude-3-Opus	0.7834	0.7839	0.8486	0.6097	0.7571	-	2	FS+Spans+Hyp
68	Claude-3-Opus	0.7666	0.7682	0.8362	0.5842	0.7403	-	3	FS+Spans+Hyp
69	Claude-3.5-Sonnet	0.8192	0.8243	0.8819	0.6392	0.8076	-	1	FS+Spans+Hyp
70	Claude-3.5-Sonnet	0.8149	0.8197	0.8792	0.6346	0.8010	-	2	FS+Spans+Hyp
71	Claude-3.5-Sonnet	0.8197	0.8246	0.8817	0.6279	0.8112	-	3	FS+Spans+Hyp
72	gemma-3-27b-it	0.7886	0.7860	0.8316	0.6348	0.7739	-	1	ZS
73	qwen2.5-72b-instruct	0.7728	0.7699	0.8248	0.5776	0.7579	-	1	ZS
74	qwen2.5-72b-instruct	0.7810	0.7754	0.8374	0.5013	0.7757	-	1	opt. ZS
75	qwen2.5-72b-instruct	0.7351	0.7241	0.8094	0.4920	0.6892	-	1	FS
76	qwen2.5-72b-instruct	0.7484	0.7432	0.8196	0.4378	0.7357	-	1	FS+Spans
77	qwen2.5-72b-instruct	0.7604	0.7505	0.8239	0.6236	0.7028	-	1	FS+Spans+Hyp
78	deepseek-chat	0.7881	0.7869	0.8496	0.6087	0.7631	-	1	ZS
79	deepseek-chat	0.7886	0.7874	0.8487	0.6139	0.7640	-	2	ZS

Table C.2: Baseline performance of models across multiple evaluation runs on the ContractNLI test set (Part 2 of 3).

No	Model ID	Acc.	F1[W]	F1[E]	F1[C]	F1[N]	Quant.	Try	Prompting
80	deepseek-chat	0.7891	0.7883	0.8477	0.6126	0.7674	-	3	ZS
81	deepseek-chat	0.7872	0.7829	0.8503	0.6128	0.7521	-	1	opt. ZS
82	deepseek-chat	0.7886	0.7843	0.8535	0.6103	0.7526	-	2	opt. ZS
83	deepseek-chat	0.7891	0.7847	0.8519	0.6154	0.7539	-	3	opt. ZS
84	deepseek-chat	0.7681	0.7607	0.8346	0.6104	0.7182	-	1	FS
85	deepseek-chat	0.7743	0.7714	0.8377	0.5812	0.7465	-	1	FS+Spans
86	deepseek-chat	0.7963	0.7939	0.8612	0.6479	0.7573	-	1	FS+Spans+Hyp
87	deepseek-reasoner	0.7398	0.7455	0.8281	0.5229	0.7112	-	1	ZS
88	deepseek-reasoner	0.7394	0.7433	0.8305	0.5270	0.7024	-	2	ZS
89	claude-3.7-Sonnet	0.7704	0.7781	0.8633	0.5602	0.7398	-	1	ZS
90	Claude-3.7-Sonnet	0.7671	0.7733	0.8546	0.5586	0.7383	-	1	opt. ZS
91	Claude-3.7-Sonnet	0.7590	0.7602	0.8463	0.5607	0.7165	-	1	FS
92	Claude-3.7-Sonnet	0.7724	0.7766	0.8538	0.5805	0.7417	-	1	FS+Spans
93	Claude-3.7-Sonnet	0.8034	0.8068	0.8746	0.6309	0.7769	-	1	FS+Spans+Hyp
94	Llama3.1-70b-Instr	0.5758	0.5462	0.5014	0.2749	0.6604	-	1	opt. ZS
95	Llama3.1-70b-Instr	0.5849	0.5559	0.5208	0.2561	0.6667	-	2	opt. ZS
96	Llama3.1-70b-Instr	0.5725	0.5417	0.5046	0.2303	0.6574	-	3	opt. ZS
97	Llama3.1-70b-Instr	0.5729	0.5506	0.5421	0.2381	0.6358	-	1	FS
98	Llama3.3-70b-Instr	0.6767	0.6716	0.7366	0.5378	0.6346	-	1	ZS
99	Llama3.3-70b-Instr	0.7164	0.7123	0.7704	0.4745	0.7080	-	1	opt. ZS
100	Llama3.3-70b-Instr	0.6657	0.6565	0.7326	0.4431	0.6268	-	1	FS
101	Llama3.3-70b-Instr	0.6915	0.6879	0.7382	0.4244	0.6982	-	1	FS+Spans
102	Llama3.3-70b-Instr	0.7102	0.6974	0.7840	0.5294	0.6455	-	1	FS+Spans+Hyp
103	Mistral-8x7b-Instr	0.5423	0.5475	0.6445	0.4103	0.4770	-	1	ZS
104	Mistral-8x7b-Instr	0.6006	0.5931	0.6717	0.1592	0.6146	-	1	opt. ZS
105	Mistral-8x7b-Instr	0.6002	0.5804	0.6836	0.1931	0.5642	-	1	FS
106	Mistral-8x7b-Instr	0.6150	0.6017	0.6901	0.1951	0.6060	-	1	FS+Spans
107	Mistral-8x7b-Instr	0.5323	0.5302	0.6494	0.4152	0.4305	-	1	FS+Spans+Hyp
108	Mistral-7b-Instr	0.5364	0.5042	0.5279	0.0248	0.5951	-	1	ZS
109	Mistral-7b-Instr	0.5084	0.4821	0.6055	0.0000	0.4672	-	1	opt. ZS
110	Mistral-7b-Instr	0.5065	0.4702	0.6053	0.0082	0.4379	-	1	FS
111	Mistral-7b-Instr	0.4940	0.4576	0.6085	0.0076	0.4053	-	1	FS+Spans
112	Mistral-7b-Instr	0.4873	0.3660	0.6460	0.0090	0.1528	-	1	FS+Spans+Hyp

Table C.3: Baseline performance of models across multiple evaluation runs on the ContractNLI test set (Part 3 of 3).

No	**Model ID**	**Acc.**	**F1[W]**	**F1[E]**	**F1[C]**	**F1[N]**	**Quant.**	**Try**	**Prompting**
0	gpt-4o	0.7328	0.7427	0.7575	0.5936	0.7630	-	1	ZS
1	gpt-4o	0.7248	0.7367	0.7598	0.5760	0.7516	-	2	ZS
2	gpt-4o	0.7303	0.7420	0.7629	0.5799	0.7593	-	3	ZS
3	gpt-4o	0.7102	0.7059	0.6685	0.6364	0.7584	-	1	opt. ZS
4	gpt-4o	0.7047	0.7008	0.6653	0.6220	0.7537	-	2	opt. ZS
5	gpt-4o	0.7012	0.6968	0.6611	0.604	0.7532	-	3	opt. ZS
6	gpt-4o	0.7353	0.7363	0.7305	0.6333	0.7658	-	1	FS
7	gpt-4o	0.7283	0.7290	0.7195	0.6341	0.7602	-	2	FS
8	gpt-4o	0.7343	0.7347	0.7222	0.6562	0.7652	-	3	FS
9	gpt-4o	0.7233	0.7255	0.7183	0.6174	0.7575	-	1	FS+Spans
10	gpt-4o	0.7328	0.7337	0.7321	0.6324	0.7587	-	2	FS+Spans
11	gpt-4o	0.7378	0.7382	0.7268	0.6534	0.7689	-	3	FS+Spans
12	gpt-4o	0.7835	0.7842	0.7908	0.7339	0.7894	-	1	CoT
13	gpt-4o	0.7835	0.7841	0.7911	0.7403	0.7874	-	2	CoT
14	gpt-4o	0.7760	0.7766	0.7824	0.7273	0.7824	-	3	CoT
15	Claude-3-Opus	0.8177	0.8194	0.8398	0.7174	0.8232	-	1	ZS
16	Claude-3-Opus	0.8217	0.8231	0.8457	0.7221	0.8246	-	2	ZS
17	Claude-3-Opus	0.8177	0.8197	0.8458	0.7024	0.8215	-	3	ZS
18	Claude-3.5-Sonnet	0.8237	0.8253	0.8450	0.7302	0.8283	-	1	ZS
19	Claude-3.5-Sonnet	0.8217	0.8233	0.8420	0.7302	0.8269	-	2	ZS
20	Claude-3.5-Sonnet	0.8237	0.8253	0.8438	0.7343	0.8285	-	3	ZS
21	Claude-3-Opus	0.8282	0.8296	0.8534	0.7164	0.8327	-	1	opt. ZS
22	Claude-3-Opus	0.8242	0.8255	0.8502	0.7110	0.8281	-	2	opt. ZS
23	Claude-3-Opus	0.8262	0.8273	0.8500	0.72	0.8301	-	3	opt. ZS
24	Claude-3.5-Sonnet	0.8378	0.8385	0.8593	0.7420	0.8408	-	1	opt. ZS
25	Claude-3.5-Sonnet	0.8312	0.8319	0.8513	0.7420	0.8339	-	2	opt. ZS
26	Claude-3.5-Sonnet	0.8408	0.8413	0.8598	0.7607	0.8421	-	3	opt. ZS
27	Claude-3-Opus	0.8378	0.8382	0.8629	0.7489	0.8349	-	1	FS
28	Claude-3-Opus	0.8338	0.8346	0.8602	0.7311	0.8337	-	2	FS
29	Claude-3-Opus	0.8302	0.8311	0.8556	0.7338	0.8298	-	3	FS
30	Claude-3.5-Sonnet	0.8398	0.8405	0.8604	0.7511	0.8419	-	1	FS
31	Claude-3.5-Sonnet	0.8438	0.8441	0.8639	0.7699	0.8420	-	2	FS
32	Claude-3.5-Sonnet	0.8413	0.8420	0.8607	0.7527	0.8446	-	3	FS
33	Claude-3-Opus	0.8348	0.8352	0.8633	0.7373	0.8306	-	1	FS+Spans
34	Claude-3-Opus	0.8343	0.8344	0.8619	0.7380	0.8301	-	2	FS+Spans
35	Claude-3-Opus	0.8277	0.8285	0.8514	0.7342	0.8281	-	3	FS+Spans
36	Claude-3.5-Sonnet	0.8348	0.8356	0.8544	0.7420	0.8391	-	1	FS+Spans
37	Claude-3.5-Sonnet	0.8413	0.8417	0.8582	0.7696	0.8424	-	2	FS+Spans
38	Claude-3.5-Sonnet	0.8428	0.8432	0.8621	0.7625	0.8435	-	3	FS+Spans
39	Claude-3-Opus	0.8378	0.8373	0.8629	0.7385	0.8354	-	1	FS+Spans+Hyp
40	Claude-3-Opus	0.8368	0.8365	0.8605	0.7466	0.8340	-	2	FS+Spans+Hyp

Table C.4: Detailed baseline performance of models across multiple evaluation runs. Evaluations are conducted on isolated spans—subsections of the contract text directly relevant to the hypothesis—. (Part 1 of 3)

No	Model ID	Acc.	F1[W]	F1[E]	F1[C]	F1[N]	Quant.	Try	Prompting
41	Claude-3-Opus	0.8358	0.8355	0.8588	0.7426	0.8344	-	3	FS+Spans+Hyp
42	Claude-3.5-Sonnet	0.8498	0.8499	0.8679	0.7919	0.8459	-	1	FS+Spans+Hyp
43	Claude-3.5-Sonnet	0.8508	0.8509	0.8644	0.8080	0.8477	-	2	FS+Spans+Hyp
44	Claude-3.5-Sonnet	0.8569	0.8569	0.8747	0.7891	0.8553	-	3	FS+Spans+Hyp
45	gemma-3-27b-it	0.8247	0.8238	0.8536	0.7378	0.8148	-	1	ZS
46	gemma-3-27b-it	0.8207	0.8183	0.8512	0.7037	0.8128	-	1	opt. ZS
47	gemma-3-27b-it	0.8127	0.8104	0.8462	0.6945	0.8025	-	1	FS
48	gemma-3-27b-it	0.8137	0.8116	0.8469	0.6939	0.8047	-	1	FS+Spans
49	gemma-3-27b-it	0.8182	0.8160	0.8472	0.7103	0.8104	-	1	FS+Spans+Hyp
50	qwen2.5-72b-instruct	0.8277	0.8263	0.8559	0.7345	0.8189	-	1	ZS
51	qwen2.5-72b-instruct	0.8217	0.8189	0.8599	0.6865	0.8098	-	1	opt. ZS
52	qwen2.5-72b-instruct	0.8212	0.8189	0.8579	0.6873	0.8116	-	1	FS
53	qwen2.5-72b-instruct	0.8227	0.8202	0.8593	0.6859	0.8134	-	1	FS+Spans
54	Qwen2.5-72b-Instr.	0.8418	0.8401	0.8651	0.7775	0.8304	-	1	FS+Spans+Hyp
55	deepseek-chat	0.8368	0.8365	0.8559	0.7692	0.8333	-	1	ZS
56	deepseek-chat	0.8373	0.8370	0.8574	0.7682	0.8330	-	2	ZS
57	deepseek-chat	0.8368	0.8366	0.8562	0.7675	0.8335	-	3	ZS
58	deepseek-chat	0.8353	0.8348	0.8564	0.7606	0.8311	-	1	opt. ZS
59	deepseek-chat	0.8348	0.8343	0.8568	0.7589	0.8299	-	2	opt. ZS
60	deepseek-chat	0.8353	0.8348	0.8573	0.7606	0.8301	-	3	opt. ZS
61	deepseek-chat	0.8307	0.8302	0.8547	0.7506	0.8250	-	1	FS
62	deepseek-chat	0.8302	0.8296	0.8593	0.7354	0.8227	-	2	FS
63	deepseek-chat	0.8368	0.8361	0.8632	0.7455	0.8309	-	3	FS
64	deepseek-chat	0.8317	0.8313	0.8571	0.7392	0.8276	-	1	FS+Spans
65	deepseek-chat	0.8368	0.8364	0.8570	0.7630	0.8335	-	2	FS+Spans
66	deepseek-chat	0.8383	0.8379	0.8637	0.7489	0.8335	-	3	FS+Spans
67	deepseek-chat	0.8473	0.8467	0.8689	0.7930	0.8375	-	1	FS+Spans+Hyp
68	deepseek-reasoner	0.7860	0.7903	0.8061	0.6528	0.8070	-	1	ZS
69	deepseek-reasoner	0.7810	0.7854	0.8011	0.6568	0.8000	-	2	ZS
70	deepseek-reasoner	0.7805	0.7852	0.8034	0.6458	0.8000	-	3	ZS
71	Claude-3.7-Sonnet	0.8061	0.8074	0.8180	0.7388	0.8131	-	1	ZS
72	Claude-3.7-Sonnet	0.8071	0.8084	0.8187	0.7403	0.8143	-	2	ZS
73	Claude-3.7-Sonnet	0.8051	0.8064	0.8162	0.7403	0.8122	-	3	ZS
74	Claude-3.7-Sonnet	0.8222	0.8230	0.8371	0.7609	0.8237	-	1	opt. ZS
75	Claude-3.7-Sonnet	0.8212	0.8220	0.8362	0.7593	0.8228	-	2	opt. ZS
76	Claude-3.7-Sonnet	0.8217	0.8225	0.8367	0.7609	0.8230	-	3	opt. ZS
77	Claude-3.7-Sonnet	0.8312	0.8326	0.8510	0.7439	0.8353	-	1	FS
78	Claude-3.7-Sonnet	0.8317	0.8327	0.8508	0.7556	0.8330	-	2	FS
79	Claude-3.7-Sonnet	0.8242	0.8252	0.8418	0.7531	0.8258	-	3	FS

Table C.5: Detailed baseline performance of models across multiple evaluation runs. Evaluations are conducted on isolated spans—subsections of the contract text directly relevant to the hypothesis—. (Part 2 of 3)

No	**Model ID**	**Acc.**	**F1[W]**	**F1[E]**	**F1[C]**	**F1[N]**	**Quant.**	**Try**	**Prompting**
80	Claude-3.7-Sonnet	0.8277	0.8288	0.8429	0.7592	0.8313	-	1	FS+Spans
81	claude-3.7-Sonnet	0.8428	0.8431	0.8601	0.7617	0.8456	-	1	FS+Spans+Hyp
82	Llama3.1-70b-Instr	0.7328	0.7321	0.7321	0.6650	0.7477	-	1	ZS
83	Llama3.1-70b-Instr	0.7332	0.7325	0.7346	0.6505	0.7496	-	2	ZS
84	Llama3.1-70b-Instr	0.7411	0.7402	0.7400	0.6650	0.7578	-	3	ZS
85	Llama3.1-70b-Instr	0.7685	0.7655	0.8045	0.5699	0.7731	-	1	opt. ZS
86	Llama3.1-70b-Instr	0.7614	0.7585	0.7934	0.5675	0.7690	-	2	opt. ZS
87	Llama3.1-70b-Instr	0.7609	0.7588	0.7895	0.5898	0.7683	-	3	opt. ZS
88	Llama3.1-70b-Instr	0.7248	0.7226	0.7372	0.5707	0.7437	-	1	FS
89	Llama3.1-70b-Instr	0.7348	0.7325	0.7517	0.5660	0.7525	-	2	FS
90	Llama3.1-70b-Instr	0.7212	0.7196	0.7306	0.5885	0.7394	-	3	FS
91	Llama3.1-70b-Instr	0.7348	0.7321	0.7380	0.5962	0.7579	-	1	FS+Spans
92	Llama3.1-70b-Instr	0.7313	0.7288	0.7449	0.5532	0.7541	-	2	FS+Spans
93	Llama3.1-70b-Instr	0.7117	0.7087	0.7081	0.5789	0.7394	-	3	FS+Spans
94	Llama3.1-70b-Instr	0.8152	0.8132	0.8441	0.6818	0.8138	-	1	FS+Spans+Hyp
95	Llama3.1-70b-Instr	0.8157	0.8138	0.8437	0.6937	0.8126	-	2	FS+Spans+Hyp
96	Llama3.1-70b-Instr	0.8192	0.8175	0.8478	0.6948	0.8167	-	3	FS+Spans+Hyp
97	Llama3.3-70b-Instr	0.7845	0.7849	0.8092	0.7089	0.7790	-	1	ZS
98	Llama3.3-70b-Instr	0.8117	0.8103	0.8389	0.7136	0.8051	-	1	opt. ZS
99	Llama3.3-70b-Instr	0.7941	0.7936	0.8177	0.6929	0.7935	-	1	FS
100	Llama3.3-70b-Instr	0.8021	0.8012	0.8303	0.6998	0.7966	-	1	FS+Spans
101	Llama3.3-70b-Instr	0.8403	0.8391	0.8651	0.7536	0.8337	-	1	FS+Spans+Hyp
102	Mistral-8x7b-Instr	0.7177	0.7192	0.7610	0.6478	0.6951	-	1	ZS
103	Mistral-8x7b-Instr	0.7515	0.7425	0.8117	0.5120	0.7290	-	1	opt. ZS
104	Mistral-8x7b-Instr	0.7574	0.7516	0.8065	0.5934	0.7350	-	1	FS
105	Mistral-8x7b-Instr	0.7438	0.7361	0.8008	0.5272	0.7217	-	1	FS+Spans
106	Mistral-8x7b-Instr	0.7514	0.7474	0.8106	0.6332	0.7125	-	1	FS+Spans+Hyp
107	Mistral-7b-Instr	0.6224	0.6052	0.6083	0.2867	0.6744	-	1	ZS
108	Mistral-7b-Instr	0.6217	0.5932	0.6828	0.0437	0.6339	-	1	opt. ZS

Table C.6: Detailed baseline performance of models across multiple evaluation runs. Evaluations are conducted on isolated spans—subsections of the contract text directly relevant to the hypothesis—. (Part 3 of 3)

Appendix D

Human Evaluation

We deployed a survey on *Prolific*, an online research platform widely used in academic studies. Prolific ensures participant anonymity and does not allow researchers to interfere with or influence responses. Participants were asked to compare answers from PAKTON and ChatGPT for the same legal question, based on specific evaluation criteria.

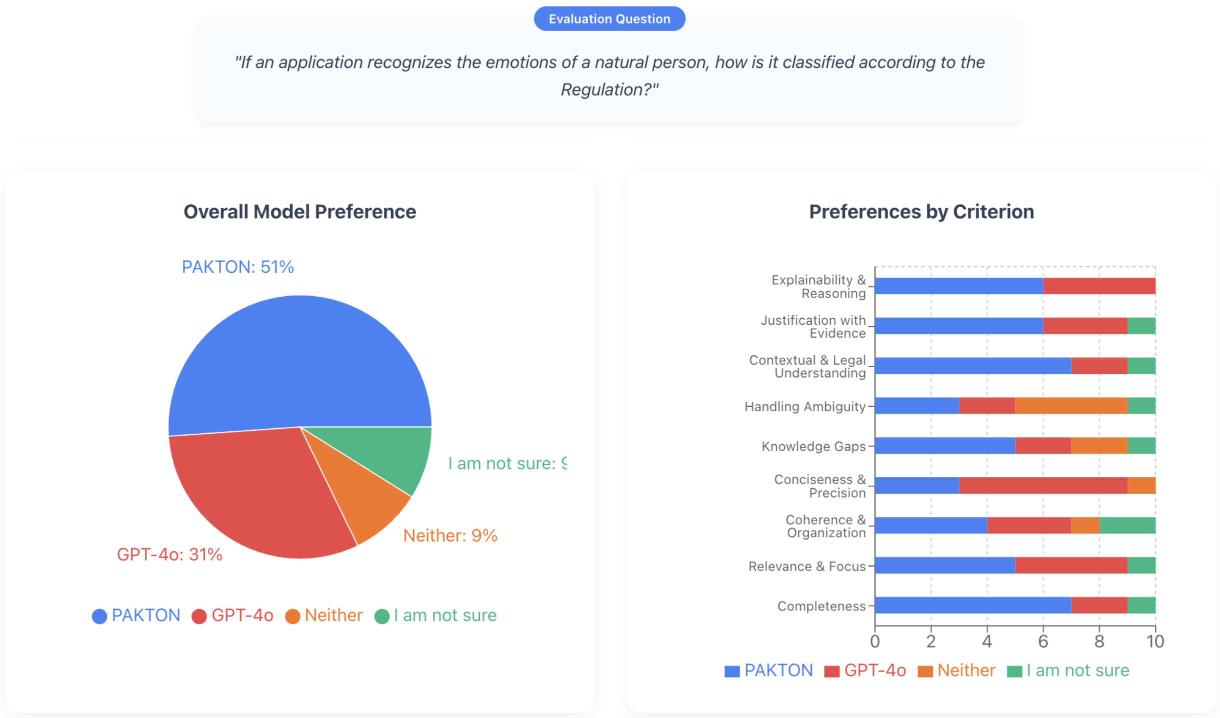
We intentionally did not restrict participation to legal professionals. Our goal was to understand how well PAKTON's responses are received by the general public, especially in comparison to a widely used baseline like ChatGPT. The only eligibility requirements were that participants must be fluent in English and have completed at least compulsory education (e.g., high school level), to ensure basic reading comprehension and critical reasoning skills.

Our human evaluation was carried out by anonymous volunteers who were fairly compensated for their time. All participants were fully informed about the purpose of the study and followed a structured annotation protocol. No personal data was collected or stored at any stage of the evaluation.

Participants were paid £9/hour, exceeding Prolific's minimum compensation rate of £6/hour. While the expected duration of the survey was 5 minutes, the median completion time was closer to 15 minutes. To fairly account for the additional effort, participants who spent more time were rewarded with bonuses, calculated based on Prolific's minimum rate.

Evaluators assessed PAKTON's outputs via the interface shown in Figure F.0.1, reviewing both predefined examples and interacting with the system in real time using their own queries. Feedback was overwhelmingly positive, highlighting the tool's practical utility.

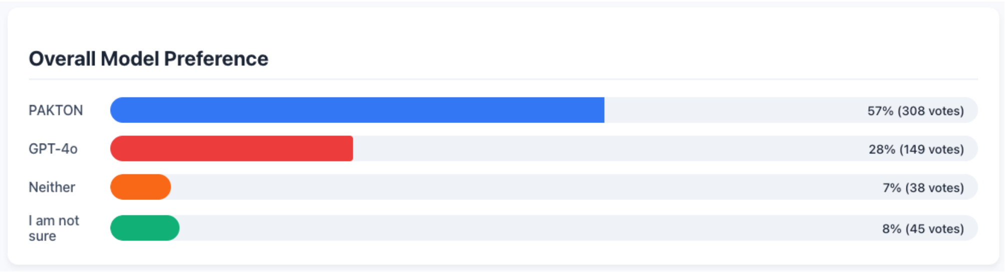
Figure D.0.1 presents a comparative assessment between PAKTON and GPT-4o, based on human judgments across multiple criteria. PAKTON consistently outperformed GPT-4o, both at the individual response level and in aggregate preferences.



Evaluation Question

"If an application recognizes the emotions of a natural person, how is it classified according to the Regulation?"

(a) Preference based on responses for a single question



(b) Overall Model Preference aggregated across all criteria and all questions

Figure D.0.1: Comparative analysis of PAKTON vs. GPT-4o based on human evaluator judgments across different criteria

Appendix E

G-EVAL Experiments

We set the temperature to 0 to ensure deterministic responses, facilitating reproducibility.

As part of our evaluation design, we ensured fair model comparison settings. For “GPT”, we implemented a (RAG) pipeline. For PAKTON, we limited tool usage to strictly in-document retrieval (disabling access to external tools like web search), and capped the number of interrogation turns at five. GPT-4o served as the underlying model for both systems to eliminate base model performance discrepancies and isolate differences due to architecture and orchestration.

The aggregated scores for all evaluated outputs of PAKTON and GPT-4o on the ContractNLI dataset are presented in Table E.1. The distribution of the aggregated scores across all evaluation criteria, as computed by the G-EVAL framework, is illustrated in Figure E.0.1.

Evaluation Criterion	**PAKTON**	**GPT**
Justification with Evidence	0.6887	0.6503
Contextual and Legal Understanding	0.8054	0.8472
Handling Ambiguity	0.6925	0.4248
Acknowledgment of Knowledge Gaps	0.9124	0.6905
Conciseness and Precision	0.8691	0.8323
Coherence and Organization	0.9024	0.5118
Relevance and Focus	0.9155	0.9030
Completeness	0.8216	0.6866
Explainability and Reasoning	0.8350	0.8127

Table E.1: Comparison of PAKTON and GPT scores acquired across different evaluation criteria aggregated for all examined outputs.

Criterion	Contextual and Legal Understanding
Score	0.87549
Reason	The report accurately interprets legal terminology and context, correctly identifies that employees are not third parties, and aligns with the contract's confidentiality obligations. It fully understands the user's query, addressing legal issues and assumptions, **but suggests further research for comprehensive understanding.**

Table E.2: Example illustrating score penalization of "Contextual and Legal Understanding" due to further research section of the report.

Criterion	**Instructions**
Explainability and Reasoning	Evaluate whether the report clearly and transparently explains not only the final conclusion, but also the reasoning process and supporting evidence in a step-by-step, understandable manner. The explanation should guide the reader through the logic in a way that supports comprehension, avoiding unexplained jumps in logic.
Justification with Evidence	Determine whether the statements and claims are explicitly justified with relevant, specific, and clearly cited evidence (e.g., direct quotations, clause references). The justification should be traceable, allowing the reader to locate the original source material.
Contextual and Legal Understanding	Assess whether the report demonstrates a deep and accurate understanding of the document, its legal terminology, and the broader context. Consider whether it correctly interprets clauses and captures implied assumptions or legal concerns behind the question.
Handling Ambiguity	Determine whether the report identifies and handles ambiguities in the source material appropriately, such as by presenting multiple interpretations or justifying a chosen one clearly.
Acknowledgment of Knowledge Gaps	Evaluate whether the report explicitly acknowledges when available information is insufficient to support a conclusion, avoiding speculation or overconfidence.
Conciseness and Precision	Assess whether the report communicates clearly and efficiently, avoiding unnecessary repetition or verbosity, while still covering all key points.
Coherence and Organization	Check whether the report is logically structured, flows smoothly, and maintains clarity across sections. Transitions between ideas should be natural and helpful.
Relevance and Focus	Evaluate whether the report stays on topic and maintains focus on answering the core question, avoiding tangents or irrelevant content.
Completeness	Assess whether the report addresses all important aspects of the question and offers a contextually broad and holistic answer. It should not omit any major points or perspectives.

Table E.3: Instructions given to human annotators for each evaluation criterion used in the PAKTON vs. ChatGPT comparison. Similar instructions were given to the G-EVAL framework.

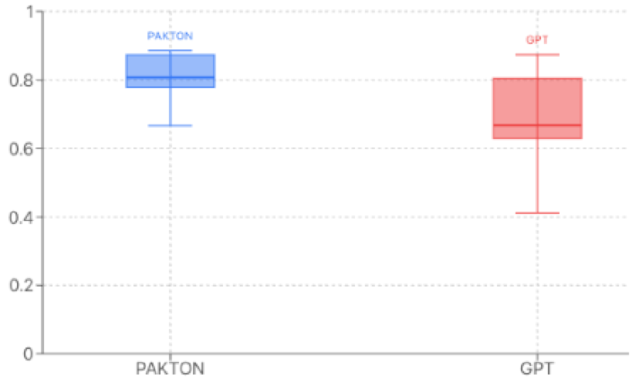


Figure E.0.1: Comparative analysis of PAKTON vs. GPT-4o based on the G-EVAL framework across different criteria. Distribution of the aggregated scores across all criteria.

Appendix F

Representative Output from PAKTON and Comparative Analysis with GPT (RAG)

The following examples illustrate the outputs generated by PAKTON and GPT (RAG) in response to the same ContractNLI input, consisting of a contract-hypothesis pair. The input is as follows:

CONTRACT NLI INPUT

Is the following hypothesis ENTAILMENT, CONTRADICTION, or NEUTRAL according to the content of the contract: <hypothesis>Receiving Party shall destroy or return some Confidential Information upon the termination of Agreement.</hypothesis>?

The problem is categorized as a Natural Language Inference (NLI) task for contract analysis. The answer can be one of the following:

ENTAILMENT:
- The hypothesis is logically true based on the content of the contract.
- It is explicitly stated or can be directly inferred.

CONTRADICTION:
- The hypothesis directly or logically conflicts with the contract's content.

NEUTRAL:
- There is insufficient information to conclude if it is CONTRADICTION or ENTAILMENT.
- If the hypothesis isn't mentioned in the contract or is unrelated.

Based on this input, GPT produced the following output:

GPT (RAG) OUTPUT

The contract does indeed entail the hypothesis that the Receiving Party shall destroy or return some Confidential Information upon the termination of the Agreement.

As outlined in section 9 of the contract, labeled “Return of JEA Confidential Information”, it specifies that “the Recipient shall immediately return and redeliver to the other all tangible material embodying the JEA Confidential Information provided hereunder. . . and all other documents or materials (’Notes’). . . upon the earlier of:
I. the completion or termination of the dealings between the parties contemplated hereunder; or
II. the termination of this Agreement; or
III. at such time as JEA may so request.”

It also provides an alternative where, with the written consent of JEA, the Recipient may destroy such information, and it provides details on how this destruction should be carried out and verified.

These obligations regarding handling Confidential Information are specific to the event of the termination of the Agreement, directly supporting the hypothesis.

While the output of PAKTON for the same input is presented as follows:

PAKTON OUTPUT

Title: Analysis of Contractual Obligations Regarding Confidential Information Post-Termination

Summary of Topic:
The legal question under consideration is whether the hypothesis that "Receiving Party shall destroy or return some Confidential Information upon the termination of Agreement" is classified as ENTAILMENT, CONTRADICTION, or NEUTRAL according to the contract’s content. This involves determining whether the contract explicitly or implicitly supports, contradicts, or is neutral regarding this hypothesis.

Legal Reasoning & Analysis:
The contract under review contains specific clauses that delineate the obligations of the Receiving Party regarding the handling of confidential information upon the termination of the agreement. The updated analysis provides a more nuanced understanding of these obligations and introduces potential exceptions and modifications:
1. Return of Confidential Information:
- The contract specifies that upon request by the disclosing party, the Receiving Party is obligated to return all confidential information.
- Exceptions exist for analyses prepared by or for the Receiving Party, and they may retain one copy for compliance with applicable laws or record retention policies [1], [2], [3].

2. Destruction of Confidential Information:
- The contract provides an option for the Receiving Party to destroy the confidential information instead of returning it, provided there is written consent from the disclosing party.
- The destruction must be verifiable, and computerized data must be non-recoverably erased [2] (Clause 9).

3. Exceptions to Obligations:
- Certain types of confidential information are exempt from the return or destruction obligations.
- This includes information received from a third party without breach, information known prior to disclosure, publicly known information, independently developed information, and information ordered for public release by a government agency [1], [3], [4], [5].

4. Retention for Legal Compliance:
- The receiving party may retain one copy of the confidential information as needed to comply with applicable law and/or the party’s record retention policies.
- This retained information must be kept subject to the terms of the agreement or destroyed [1], [2], [3].
- The retention is allowed to enable compliance with statutory document retention policies and must be necessary for legal compliance, suggesting that any retention beyond this scope would not be permissible [4].

5. Impact of Legal Compliance and Record Retention Policies:
- The contract explicitly allows retention for compliance with legal and record retention requirements, thereby modifying the general obligation to return or destroy the information upon termination.
- This impacts the classification of the hypothesis as ENTAILMENT, as the contract articulates exceptions to the return or destruction requirement for compliance purposes [1], [2], [3].

Preliminary Answer & Direction for Further Research:
The hypothesis that the Receiving Party shall destroy or return some Confidential Information upon termination appears to be **ENTAILMENT** with specific conditions and exceptions.
The contract outlines obligations for returning or destroying confidential information, with allowances for retention due to legal compliance and record retention policies.
The determination of ENTAILMENT assumes that the specific conditions outlined in the contract are met, such as the disclosing party's request or consent and the applicability of exceptions for certain types of information.
Further research should focus on verifying whether these conditions apply universally to all confidential information or are subject to specific circumstances or exceptions not covered in the provided excerpts.

Gaps & Next Questions:
- Clarification on Universal Application: Is the obligation to return or destroy confidential information upon termination applicable to all types of confidential information, or are there exceptions not covered in the provided clauses?

...

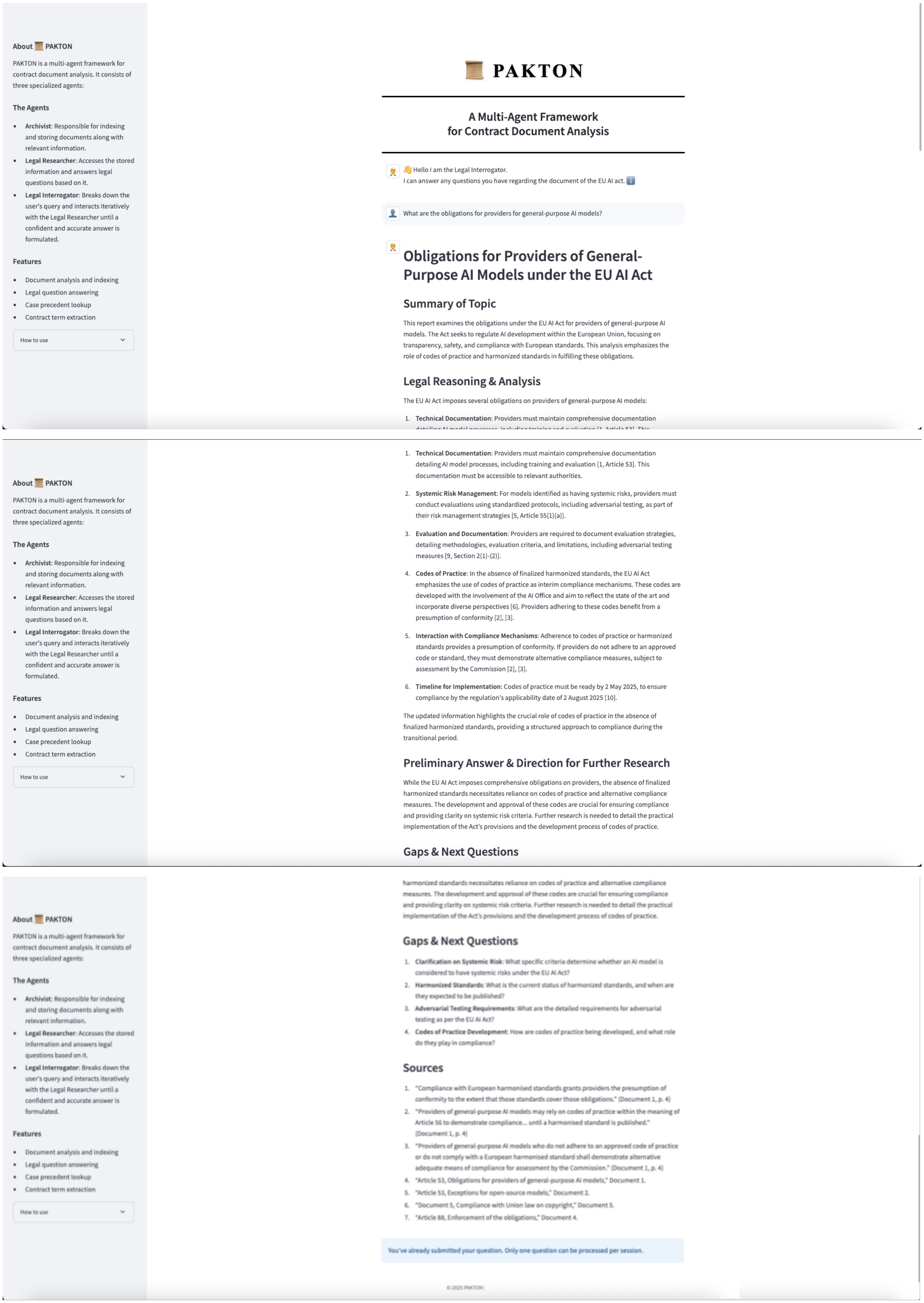
- Impact of Legal Compliance: How does the retention of information for legal compliance or record retention policies affect the overall obligations regarding confidential information?

Sources:
1. Clause 5 - Limits on Confidential Information
2. Clause 4.e - Obligations of Receiving Party
3. Clause 4.e - Obligations of Receiving Party
4. Clause 5 - Limits on Confidential Information
5. Clause 5.e - Limits on Confidential Information

A comparison of the outputs generated by GPT and PAKTON in response to the same ContractNLI input reveals substantial differences in terms of depth, structure, and legal interpretability. The output from GPT is concise and directly addresses the hypothesis by selecting the label *ENTAILMENT*. It supports this classification by referencing specific contractual clauses that align with the hypothesis and briefly justifies the decision. While this approach is effective for providing a quick assessment, the explanation remains relatively shallow. It does not explore potential exceptions, conditional obligations, or broader legal implications that may influence the interpretation.

On the other hand, the output from **PAKTON** exhibits a much more detailed and structured analysis. It decomposes the legal obligations regarding confidential information into separate categories: return, destruction, exceptions, and legal compliance. Each element is examined thoroughly, with references to specific clauses and relevant conditions. PAKTON also integrates legal reasoning, raises critical questions for further investigation, and acknowledges limitations or ambiguities in the contract's language. Additionally, PAKTON considers practical legal factors such as retention for statutory compliance and record-keeping policies—factors which are absent from GPT's output. These aspects contribute to a more nuanced and legally grounded evaluation of the hypothesis.

Overall, PAKTON's response reflects a deeper engagement with the legal content, demonstrating an interpretive capacity closer to that of a legal expert. GPT, while effective in recognizing textual entailment, remains more limited in its reasoning and scope of analysis.



About PAKTON

PAKTON is a multi-agent framework for contract document analysis. It consists of three specialized agents:

The Agents

- **Archivist**: Responsible for indexing and storing documents along with relevant information.
- **Legal Researcher**: Accesses the stored information and answers legal questions based on it.
- **Legal Interrogator**: Breaks down the user's query and interacts iteratively with the Legal Researcher until a confident and accurate answer is formulated.

Features

- Document analysis and indexing
- Legal question answering
- Case precedent lookup
- Contract term extraction

PAKTON

A Multi-Agent Framework
for Contract Document Analysis

👋 Hello I am the Legal Interrogator.
I can answer any questions you have regarding the document of the EU AI act. ℹ️

What are the obligations for providers for general-purpose AI models?

Obligations for Providers of General-Purpose AI Models under the EU AI Act

Summary of Topic

This report examines the obligations under the EU AI Act for providers of general-purpose AI models. The Act seeks to regulate AI development within the European Union, focusing on transparency, safety, and compliance with European standards. This analysis emphasizes the role of codes of practice and harmonized standards in fulfilling these obligations.

Legal Reasoning & Analysis

The EU AI Act imposes several obligations on providers of general-purpose AI models:

1. **Technical Documentation**: Providers must maintain comprehensive documentation

About PAKTON

PAKTON is a multi-agent framework for contract document analysis. It consists of three specialized agents:

The Agents

- **Archivist**: Responsible for indexing and storing documents along with relevant information.
- **Legal Researcher**: Accesses the stored information and answers legal questions based on it.
- **Legal Interrogator**: Breaks down the user's query and interacts iteratively with the Legal Researcher until a confident and accurate answer is formulated.

Features

- Document analysis and indexing
- Legal question answering
- Case precedent lookup
- Contract term extraction

How to use

1. **Technical Documentation**: Providers must maintain comprehensive documentation detailing AI model processes, including training and evaluation [1, Article 53]. This documentation must be accessible to relevant authorities.

2. **Systemic Risk Management**: For models identified as having systemic risks, providers must conduct evaluations using standardized protocols, including adversarial testing, as part of their risk management strategies [5, Article 55(1)(a)].

3. **Evaluation and Documentation**: Providers are required to document evaluation strategies, detailing methodologies, evaluation criteria, and limitations, including adversarial testing measures [9, Section 2(1)-(2)].

4. **Codes of Practice**: In the absence of finalized harmonized standards, the EU AI Act emphasizes the use of codes of practice as interim compliance mechanisms. These codes are developed with the involvement of the AI Office and aim to reflect the state of the art and incorporate diverse perspectives [6]. Providers adhering to these codes benefit from a presumption of conformity [2], [3].

5. **Interaction with Compliance Mechanisms**: Adherence to codes of practice or harmonized standards provides a presumption of conformity. If providers do not adhere to an approved code or standard, they must demonstrate alternative compliance measures, subject to assessment by the Commission [2], [3].

6. **Timeline for Implementation**: Codes of practice must be ready by 2 May 2025, to ensure compliance by the regulation's applicability date of 2 August 2025 [10].

The updated information highlights the crucial role of codes of practice in the absence of finalized harmonized standards, providing a structured approach to compliance during the transitional period.

Preliminary Answer & Direction for Further Research

While the EU AI Act imposes comprehensive obligations on providers, the absence of finalized harmonized standards necessitates reliance on codes of practice and alternative compliance measures. The development and approval of these codes are crucial for ensuring compliance and providing clarity on systemic risk criteria. Further research is needed to detail the practical implementation of the Act's provisions and the development process of codes of practice.

Gaps & Next Questions

About PAKTON

PAKTON is a multi-agent framework for contract document analysis. It consists of three specialized agents:

The Agents

- **Archivist**: Responsible for indexing and storing documents along with relevant information.
- **Legal Researcher**: Accesses the stored information and answers legal questions based on it.
- **Legal Interrogator**: Breaks down the user's query and interacts iteratively with the Legal Researcher until a confident and accurate answer is formulated.

Features

- Document analysis and indexing
- Legal question answering
- Case precedent lookup
- Contract term extraction

How to use

harmonized standards necessitates reliance on codes of practice and alternative compliance measures. The development and approval of these codes are crucial for ensuring compliance and providing clarity on systemic risk criteria. Further research is needed to detail the practical implementation of the Act's provisions and the development process of codes of practice.

Gaps & Next Questions

1. **Clarification on Systemic Risk**: What specific criteria determine whether an AI model is considered to have systemic risks under the EU AI Act?
2. **Harmonized Standards**: What is the current status of harmonized standards, and when are they expected to be published?
3. **Adversarial Testing Requirements**: What are the detailed requirements for adversarial testing as per the EU AI Act?
4. **Codes of Practice Development**: How are codes of practice being developed, and what role do they play in compliance?

Sources

1. "Compliance with European harmonised standards grants providers the presumption of conformity to the extent that those standards cover those obligations." (Document 1, p. 4)
2. "Providers of general-purpose AI models may rely on codes of practice within the meaning of Article 56 to demonstrate compliance... until a harmonised standard is published." (Document 1, p. 4)
3. "Providers of general-purpose AI models who do not adhere to an approved code of practice or do not comply with a European harmonised standard shall demonstrate alternative adequate means of compliance for assessment by the Commission." (Document 1, p. 4)
4. "Article 53, Obligations for providers of general-purpose AI models," Document 1.
5. "Article 53, Exceptions for open-source models," Document 2.
6. "Document 5, Compliance with Union law on copyright," Document 5.
7. "Article 88, Enforcement of the obligations," Document 4.

You've already submitted your question. Only one question can be processed per session.

© 2025 PAKTON

Figure F.0.1: The user interface (UI) of PAKTON employed during the human evaluation with study participants.