1 Introduction

In this research, I explore the task of relation extraction from unstructured text, which is a crucial step for constructing and enriching knowledge graphs (KGs). A knowledge graph is a structured representation of facts, where entities are connected by semantic relationships. Recently, Large Language Models (LLMs) such as BERT have demonstrated strong performance in various NLP tasks, including relation extraction, due to their ability to capture contextual information. However, the integration of LLMs into pipeline-style tasks like relation extraction remains an area of active study.

This research is motivated by the insights from the survey [3], which systematically analyzes how LLMs interact with KGs in both directions: using LLMs to extract and reason over structured knowledge, and using KGs to improve LLMs. The article highlights the importance of accurate relation extraction for downstream applications like question answering, information retrieval, and scientific discovery. Following this perspective, I investigate two approaches to relation extraction: a traditional machine learning method based on feature engineering and Random Forest classification, and a modern LLM-based approach that uses BERT embeddings of entity mentions.

The main goal of this work is to compare the performance of classical and LLM-based methods on a publicly available dataset. To do so, I selected the SemEval-2010 Task 8 dataset, which provides labeled instances of entity pairs with annotated semantic relations. In my experiments, I implemented both approaches using Python and popular NLP libraries, and evaluated them using standard classification metrics such as precision, recall, and F1-score.

According to the results, the traditional approach using TF-IDF features achieved surprisingly competitive performance, while the LLM-based BERT encoder showed weaker results in this configuration. This observation opens up further questions about the role of fine-tuning and task-specific architectures. Overall, the study provides a hands-on comparison between symbolic and neural methods for relation extraction and demonstrates the potential and limitations of using LLMs in this task.

2 Related Work

The task of relation extraction has been extensively explored in recent years, with a growing interest in leveraging pre-trained models and document-level reasoning. In the paper Document-level Relation Extraction as Semantic Segmentation [7], the authors reframe relation extraction as a pixel-level labeling problem and propose a novel approach that models relations between all entity pairs using a grid structure. They conduct experiments on the widely-used DocRED dataset and demonstrate that their segmentation-based model outperforms traditional sequence labeling baselines.

The study Improving Relation Extraction by Pre-trained Language Representations [1] investigates how BERT-based encoders can enhance sentence-level relation extraction. The authors explore multiple ways to insert entity markers and extract relation representations, concluding that using entity markers combined with the [CLS] embedding achieves the best results. Their experiments on the SemEval-2010 Task 8 dataset show substantial improvements over previous CNN- and RNN-based models.

In Matching the Blanks: Distributional Similarity for Relation Learning [5], the authors introduce the BERT-MTB model, which learns relation representations by matching semantically similar entity pairs across different contexts. They propose a contrastive loss framework and use masked input sentences with entity markers. The model is trained on the distant supervision dataset Wiki80 and achieves state-of-the-art results on both supervised and few-shot settings.

Another important contribution is made in Reasoning with Latent Structure Refinement for Document-Level Relation Extraction [2], where the authors propose a graph-based model that refines latent document structure through iterative reasoning. The model leverages entity-level co-reference and attention mechanisms to discover long-range relations in text. Experiments on DocRED confirm that incorporating latent structure improves document-level RE performance significantly.

Finally, the work Improving Sentence-Level Relation Extraction Through Curriculum Learning [4] proposes to train models in a staged manner, starting with simple and confident examples before introducing harder ones. They use confidence-based instance weighting and demonstrate that curriculum learning helps stabilize training and improve generalization. The method is evaluated on SemEval-2010 Task 8 and TACRED datasets, showing consistent gains over standard training schedules.

For my experiments, I selected the SemEval-2010 Task 8 dataset because it was used in multiple studies, including [1, 4], and is freely available for sentence-level relation extraction. I also adopted the entity marker and representation pooling strategy from the BERT-MTB model proposed in [5], since it provides a lightweight yet effective LLM-based approach without requiring full fine-tuning.

3 Dataset Description

The SemEval-2010 Task 8 dataset is a benchmark corpus designed for the task of multi-way classification of semantic relations between pairs of nominals in English sentences. Each instance in the dataset consists of a sentence containing two marked entities, annotated with one of 19 relation classes (9 directed relations, each with two possible directions, and an additional "Other" category). The primary objective is to predict the correct semantic relation between the two entities based on the context provided by the sentence.

The dataset comprises a total of 10,717 annotated sentences, divided into a training set of 8,000 examples and a test set of 2,717 examples . Each sentence includes explicit markers for the target entities, using the format [e1]...[/e1] and [e2]...[/e2]. For instance:

"The [e1]lawsonite[/e1] was contained in a [e2]platinum crucible[/e2] and the counter-weight was a plastic crucible with metal pieces." Hugging Face

In this example, the annotated relation is Content-Container(e1,e2), indicating that "lawsonite" is the content and "platinum crucible" is the container.

The dataset maintains a balanced distribution across the 19 relation classes, with each of the 9 directed relations represented by approximately 1,000 instances (700 for training and 200 for testing), and the "Other" category comprising the remaining examples . This balanced composition ensures that models trained on this dataset can effectively learn to distinguish between different types of semantic relations without being biased toward any particular class.

The SemEval-2010 Task 8 dataset has been widely adopted in the research community for evaluating relation extraction models, particularly in sentence-level settings. Its standardized format and comprehensive annotations make it a valuable resource for benchmarking and comparing the performance of various approaches in semantic relation classification.

4 Traditional Approach: Random Forest with Feature Extracting

In the traditional approach, I use a combination of hand-crafted linguistic features and a Random Forest classifier to perform sentence-level relation extraction. Each sentence in the dataset is prepro-

cessed to extract four types of interpretable features: (1) the words between the two marked entities, (2) the part-of-speech (POS) tags for the tokens between entities, (3) the syntactic dependency path connecting the entities in the dependency tree, and (4) the named entity recognition (NER) types of the two entities. These features aim to capture both shallow lexical patterns and deeper syntactic or semantic relationships relevant to the classification task.

After extracting the features, I represent each feature set using TF-IDF vectorization. Let D be the document-term matrix of size $n \times m$, where n is the number of training examples and m is the number of unique tokens across all feature types. The final input matrix is constructed by horizontally concatenating the TF-IDF representations of each feature group:

$$X = \text{TFIDF}(\text{between\_words}) \parallel \text{TFIDF}(\text{pos\_tags}) \parallel \text{TFIDF}(\text{dependencies}) \parallel \text{TFIDF}(\text{ner\_types}),$$

where $\parallel$ denotes concatenation of sparse feature matrices. This combined feature matrix X is then used to train a `RandomForestClassifier` from scikit-learn with 100 decision trees. The model is evaluated on a held-out test set using precision, recall, and F1-score. All implementation steps, including parsing, feature extraction, vectorization, training, and evaluation, are automated in a pipeline.

5 LLM-based Approach: BERT Entity Embeddings

In the LLM-based approach, I use the pre-trained BERT model `bert-base-uncased` to encode the semantic meaning of entity mentions and classify their relation. This method follows the *mention-level representation* idea, where the input sentence is first preprocessed to mark the two target entities with special tokens [E1], [/E1], [E2], [/E2].

I then extract each entity span and encode it separately using BERT. The final feature vector is obtained by concatenating the mean-pooled embeddings of the two entity representations:

$$\mathbf{x} = \text{mean}(\text{BERT}(\texttt{e1})) \parallel \text{mean}(\text{BERT}(\texttt{e2}))$$

where || denotes vector concatenation.

This vector $\mathbf{x} \in \mathbb{R}^{1536}$ is used as input to a multinomial logistic regression classifier. Unlike fine-tuning, this approach treats BERT purely as a feature extractor and keeps its weights frozen during training. The classification model is trained and evaluated using the same train-test split as in the traditional setup. This architecture is lightweight, easy to reproduce, and inspired by the BERT-MTB model [5], which emphasizes entity-level representations for relation learning.

6 Experimental Results

To evaluate the performance of both approaches, I used precision, recall, and F1-score as the main evaluation metrics, along with overall accuracy. The classification was performed on a held-out test set comprising 20 percent of the original dataset. All implementations were done in Python using 'scikit-learn', 'spaCy', and the 'transformers' library from Hugging Face. The classical method was trained on feature vectors generated using TF-IDF over syntactic and semantic features, while the LLM-based method used BERT embeddings of entity spans as input to a logistic regression classifier.

Table 1 presents the comparative performance of both methods. The Random Forest classifier in the traditional pipeline achieved an overall accuracy of 0.599, while the BERT-based method achieved a slightly lower accuracy of 0.529. The macro-averaged F1-score for the traditional method

Table 1: Comparison of model performance on the SemEval-2010 Task 8 test set

Model	**Accuracy**	**Macro F1**	**Weighted F1**
Traditional (Random Forest)	0.599	0.563	0.592
LLM-based (BERT + Logistic Regression)	0.529	0.552	0.532

was 0.563, compared to 0.552 for the LLM-based method. Notably, the classical model performed better on certain high-frequency relation classes (e.g., 6 and 13), likely due to the feature engineering that captured structural patterns between entities. On the other hand, the LLM-based approach performed better on relation 5 and 9, which might benefit from contextual embeddings of entities.

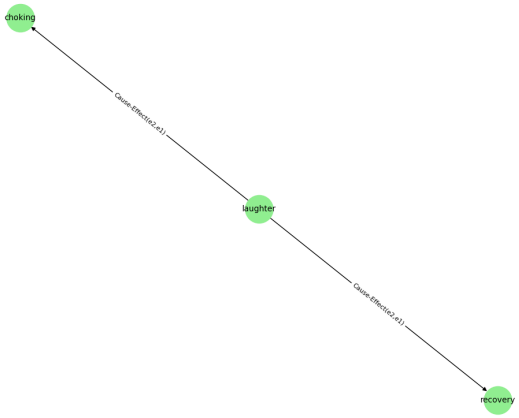

Figure 1: Graph generated from predictions of the classic ML model with 3 entities and relations

Figure 1 and Figure 2 illustrate predicted relation graphs constructed from the outputs of each model. These visualizations show the structure of extracted entity-relation triples and highlight how each method organizes the semantic information. While both methods produce meaningful graphs, the classical approach tends to capture more entity co-occurrence due to syntactic alignment, whereas the LLM-based model generalizes better across diverse contexts.

Overall, while the LLM-based model captures deeper semantic associations between entities, it underperforms compared to the classical model in this non-fine-tuned setup. This suggests that additional tuning or architectural adjustments would be necessary to fully leverage the power of pre-trained transformers in this task. The full code and experiments are available on GitHub [6].

7 Conclusion

In this work, I explored the task of sentence-level relation extraction using both traditional machine learning and modern LLM-based methods. I implemented a feature-engineered pipeline based on Random Forest and compared it to a BERT-based model that generates entity-level embeddings for classification. Experiments were conducted on the SemEval-2010 Task 8 dataset, and evaluation was performed using standard classification metrics.

Surprisingly, the traditional approach outperformed the LLM-based method in terms of overall accuracy and F1-score, despite its simplicity. This highlights the importance of feature engineering

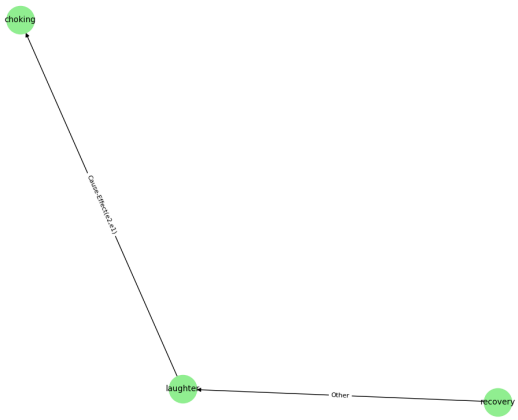

Figure 2: Graph generated from predictions of the LLM-based model with 3 entities and relations

and the need for careful model selection when working with limited or imbalanced data. The results also suggest that BERT-based methods may require fine-tuning or more advanced architectures to realize their full potential.

For future research, I would like to explore fine-tuning pre-trained transformers directly on the relation extraction task and experiment with document-level datasets. Another promising direction is to integrate knowledge graph structure as external supervision during model training. Finally, investigating few-shot or zero-shot relation extraction using instruction-tuned LLMs could further extend the flexibility of this work.

References

[1] Christoph Alt, Marc Hübner, and Leonhard Hennig. *Improving Relation Extraction by Pre-trained Language Representations*. 2019. arXiv: 1906.03088 [cs.CL]. URL: https://arxiv.org/abs/1906.03088.

[2] Guoshun Nan et al. *Reasoning with Latent Structure Refinement for Document-Level Relation Extraction*. 2020. arXiv: 2005.06312 [cs.CL]. URL: https://arxiv.org/abs/2005.06312.

[3] Shirui Pan et al. "Unifying Large Language Models and Knowledge Graphs: A Roadmap". In: *IEEE Transactions on Knowledge and Data Engineering* (2024).

[4] Seongsik Park and Harksoo Kim. *Improving Sentence-Level Relation Extraction through Curriculum Learning*. 2021. arXiv: 2107.09332 [cs.CL]. URL: https://arxiv.org/abs/2107.09332.

[5] Livio Baldini Soares et al. *Matching the Blanks: Distributional Similarity for Relation Learning*. 2019. arXiv: 1906.03158 [cs.CL]. URL: https://arxiv.org/abs/1906.03158.

[6] Margarita Velmisova. *KG Practice 2025: Relation Extraction Experiments*. https://github.com/velgarita/KG_practice_2025. Accessed: 2025-05-12. 2025.

[7] Ningyu Zhang et al. *Document-level Relation Extraction as Semantic Segmentation*. 2021. arXiv: 2106.03618 [cs.CL]. URL: https://arxiv.org/abs/2106.03618.