UNIVERSITÀ
DEGLI STUDI
DI MILANO

Università degli Studi di Milano

Text Mining and Sentiment Analysis

KG+LLM: A Happy Marriage

Author: Emil Soltanov

January 2025

Abstract

This report explores the integration of large language models (LLMs) with knowledge graph (KG) construction and reasoning tasks. By leveraging models such as LSTM, BiLSTM, BERT, and RoBERTa, we aim to streamline the KG creation process and advance knowledge representation. The project analyzes performance metrics, training history, and visualizations to highlight the capabilities and limitations of each approach in tasks like relation classification and knowledge graph visualization.

Contents

List of Figures

KG+LLM: A Happy Marriage

1 Introduction

1.1 Background

Knowledge graphs (KGs) are structured ways of representing information, where entities like people, places, or organizations are connected by meaningful relationships, such as `works_at` or `located_in`. KGs are widely used in various areas like search engines, recommendation systems, and intelligent virtual assistants because they help combine and analyze data from diverse sources. By linking different pieces of information, KGs create a clear and comprehensive picture of complex relationships, enabling deeper insights.

Historically, building knowledge graphs has been a labor-intensive process, heavily reliant on manual efforts. Domain experts have had to painstakingly extract facts from unstructured data, connect those facts, and organize them into graph structures. However, with the rapid growth of data, these manual methods have become impractical, struggling to keep up with the sheer volume and diversity of information available today. This has driven the need to explore automated ways of constructing KGs, particularly through natural language processing (NLP) techniques.

1.2 Problem Statement

In recent years, advancements in NLP, especially transformer-based models like BERT and RoBERTa, have shown great promise in tasks such as entity recognition, relation extraction, and text classification. These models can process large amounts of textual data, identify meaningful patterns, and extract relationships with high accuracy. However, integrating these powerful tools into the KG-building process remains a relatively untapped area.

The main challenge lies in using these models not just to extract relationships but also to organize them in a way that enhances the usability and scalability of KGs. Additionally, understanding the trade-offs—such as computational cost versus accuracy—of different models is vital for tailoring solutions to specific use cases.

1.3 Objectives

This project seeks to tackle these challenges by investigating how machine learning models can be integrated into the process of constructing KGs. The core objectives include:

1. **Performance Evaluation**: Analyze the performance of models such as LSTM, BiLSTM, BERT, and RoBERTa on relation classification tasks, using metrics like accuracy, F1-score, and loss convergence.

- Relevant metrics:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Predictions}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

2. **Knowledge Graph Construction**: Use the relationships identified by these models to build a KG, and then visualize and examine its structure. Relevant plots to include:
   - Distribution of relationship types across the dataset
   - Visualization of the complete Knowledge Graph structure
3. **Insights and Limitations**: Compare the strengths and weaknesses of each model, focusing on factors like computational efficiency, scalability, and how they handle imbalanced datasets.

By achieving these goals, this project aims to shed light on how to automate KG construction more effectively while providing practical insights into the trade-offs between different machine learning models.

2 Dataset

2.1 Description

The dataset for this project is specifically tailored for relation classification tasks and the construction of knowledge graphs. It contains annotated data that defines semantic connections between entities. The dataset focuses on the following types of relationships:

- `date_of_birth`: Indicates the birth year or exact date of a person.
- `place_of_birth`: Identifies the geographical location where an individual was born.
- `place_of_death`: Refers to the location where an individual passed away.
- `institution`: Links individuals to organizations such as workplaces or universities.
- `degree`: Captures academic qualifications earned by individuals.

These relationship types are widely applicable in real-world scenarios, making this dataset an excellent test bed for evaluating how well machine learning models can classify diverse relations.

2.2 Structure

Each record in the dataset includes the following elements:

- **Relation**: The type of connection between two entities.
- **Subject-Object Pair**: The two entities involved in the relationship. For example:
  - Subject: James Cunningham
  - Object: 1973
  - Relation: `date_of_birth`
- **Evidence**: Supporting textual data or references (such as URLs) that validate the relationship.

This structured format ensures the consistency needed for training, evaluation, and eventual construction of the knowledge graph.

2.3 Preprocessing

Before training the models, the dataset underwent several preprocessing steps to address common challenges like class imbalance and unstructured data. These steps included:

Normalization

- **Purpose**: Ensure that entities and relationships are represented in a standard format.
- **Example**: Convert variations of the same location, such as "New York City" and "NYC," into a single standardized format.

Balancing

- **Purpose**: Handle the issue of class imbalance, where some relationships (like `institution`) are overrepresented while others (like `place_of_birth`) are underrepresented.
- **Techniques Used**:
  - **Oversampling**: Duplicate examples from underrepresented classes to increase their presence in the dataset.
  - **Undersampling**: Reduce the number of examples from overrepresented classes to balance the dataset.

Tokenization

- **Sequence Models (LSTM, BiLSTM)**: Text was tokenized into sequences of word indices based on a predefined vocabulary.
- **Transformer Models (BERT, RoBERTa)**: Text was broken down into subword tokens using pretrained tokenizers, which help preserve the underlying meaning.

Output Labels Relationships were mapped to numerical labels for training purposes. For example:

- `date_of_birth` → 0
- `place_of_birth` → 1
- `place_of_death` → 2
- `institution` → 3
- `degree` → 4

2.4 Dataset Insights

Here are some key observations about the dataset:

- **Distribution**: There was a noticeable imbalance in the representation of different relations, which made balancing techniques essential to ensure fair model performance.
- **Scale**: The dataset contained thousands of annotated examples, providing sufficient data for training and evaluation.
- **Quality**: Each relationship came with evidence, ensuring a solid foundation for validating the extracted connections.

Relevant plot for relationship distribution:

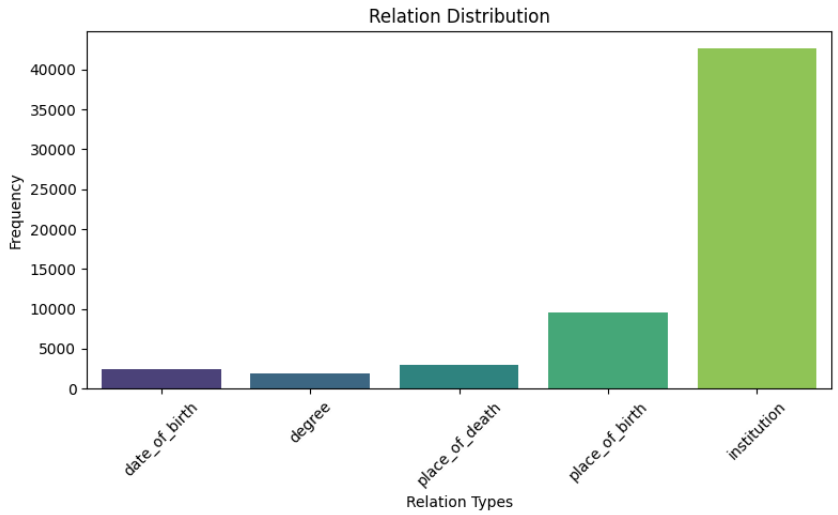

Figure 1: Distribution of the Relationship Types.

3 Methodology

3.1 Models

To evaluate performance on relation classification tasks, this project used four machine learning models: two sequence-based architectures (LSTM and BiLSTM) and two

transformer-based architectures (BERT and RoBERTa). Each model was selected for its unique strengths and capabilities.

3.1.1 LSTM

- **Overview**: Long Short-Term Memory (LSTM) networks are designed to capture long-term dependencies in sequential data, making them suitable for text-based tasks.
- **Configuration**:
  - Embedding dimensions: 256
  - Hidden dimensions: 256
  - Layers: 2
  - Dropout: 0.5
  - Learning rate: 0.0001
- **Strengths**: LSTM is simple and computationally efficient, especially when dealing with smaller datasets.
- **Limitations**: It struggles with more complex relationships and does not perform as well on large-scale datasets compared to transformer-based models.

3.1.2 BiLSTM

- **Overview**: Bidirectional LSTM (BiLSTM) networks improve on standard LSTM by processing text in both forward and backward directions, providing richer context.
- **Configuration**:
  - Embedding dimensions: 256
  - Hidden dimensions: 512
  - Layers: 2
  - Dropout: 0.5
  - Learning rate: 5e-05
- **Strengths**: BiLSTM captures semantic nuances more effectively, thanks to its bidirectional architecture.
- **Limitations**: It has a higher computational cost compared to LSTM and still falls short of transformer-based models in overall performance.

3.1.3 BERT

- **Overview**: Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based model pretrained on large amounts of text. It excels at understanding both local and global context within text.
- **Configuration**:

  - Pretrained model: `bert-base-uncased`
  - Dropout: 0.4
  - Learning rate: 1e-05

- **Strengths**: BERT delivers high accuracy and strong generalization across various relations, making it highly reliable.
- **Limitations**: It is computationally expensive and requires significant memory, making it less practical for resource-constrained environments.

3.1.4 RoBERTa

- **Overview**: RoBERTa (Robustly Optimized BERT Approach) is a variant of BERT designed to improve performance through larger training datasets and the removal of next-sentence prediction tasks.
- **Configuration**:
  - Pretrained model: `roberta-base`
  - Dropout: 0.5
  - Learning rate: 3e-06
- **Strengths**: RoBERTa is robust against overfitting, thanks to its enhanced training strategies, and performs well on diverse tasks.
- **Limitations**: It requires even more computational resources than BERT, partly due to its extended training process.

3.2 Training and Evaluation

A consistent framework was applied to train and evaluate each model, ensuring a fair comparison of their performance.

3.2.1 Metrics

The following metrics were used to assess the models' effectiveness, along with their mathematical definitions:

- **Accuracy**:
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Predictions}}$$
Accuracy measures the overall correctness of the model by dividing the total number of correct predictions (True Positives and True Negatives) by the total number of predictions.
- **Precision**:
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
Precision evaluates how many of the predicted positive cases are actually true positives. A high precision indicates a low rate of false positives.

- **Recall**:
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$
Recall measures the model's ability to identify all relevant instances (true positives) out of the total actual positives. A high recall indicates a low rate of false negatives.

- **F1-Score**:
$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$
The F1-Score is the harmonic mean of precision and recall, providing a single metric that balances both. It is particularly useful for imbalanced datasets.

- **Loss**: Cross-entropy loss was used to optimize the models during training:
$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \cdot \log \hat{y}_{i,c}$$
Here, N is the number of samples, C is the number of classes, $y_{i,c}$ is the true label (one-hot encoded), and $\hat{y}_{i,c}$ is the predicted probability for class c. Cross-entropy loss penalizes incorrect predictions, helping the model learn the correct class distributions.

3.2.2 Tools and Libraries

The project utilized several tools and libraries to streamline training and evaluation:

- **PyTorch**: For building and training neural networks.
- **Transformers (Hugging Face)**: Pretrained models and tokenizers for BERT and RoBERTa.
- **Pandas and Matplotlib**: For data manipulation and visualization.

3.2.3 Training Workflow

The training process followed these steps:

1. **Data Splitting**: The dataset was divided into training, validation, and test sets while maintaining class distribution across splits.
2. **Model Training**:
    - For sequence models (LSTM, BiLSTM), text was tokenized into sequences of word indices.
    - For transformer models (BERT, RoBERTa), tokenized text and pretrained embeddings were used as input.
3. **Hyperparameter Tuning**: Parameters such as dropout rates, learning rates, and hidden dimensions were fine-tuned to optimize performance and prevent overfitting.
4. **Evaluation**: The models were tested on the test set, with results analyzed using classification reports and confusion matrices. Training and validation loss curves were reviewed to ensure stable learning and convergence.

4 Results

4.1 Model Performance

Each model's performance was evaluated based on metrics such as accuracy, F1-scores, and class-specific metrics. Below is a summary of the results for each model:

4.1.1 LSTM

- **Overall Performance**:
  - Accuracy: 74%
  - Macro F1-Score: 0.73
- **Strengths**:
  - Performed exceptionally well on the `degree` relation (F1 = 1.00), showing its capability in handling simpler and well-represented classes.
- **Weaknesses**:
  - Struggled with the `place_of_birth` relation (Recall = 0.34), indicating challenges in identifying less frequent or ambiguous relationships.

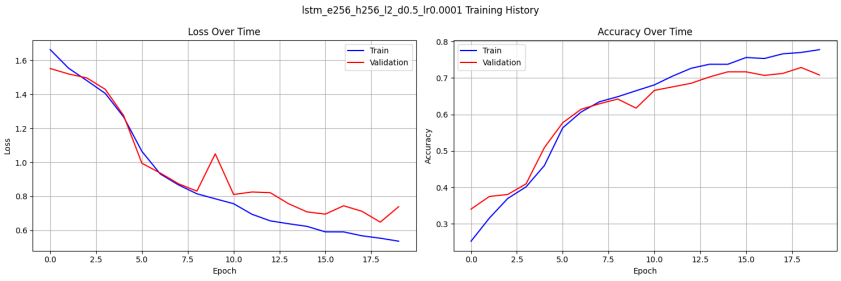

Figure 2: Training and validation loss and accuracy trends for LSTM.

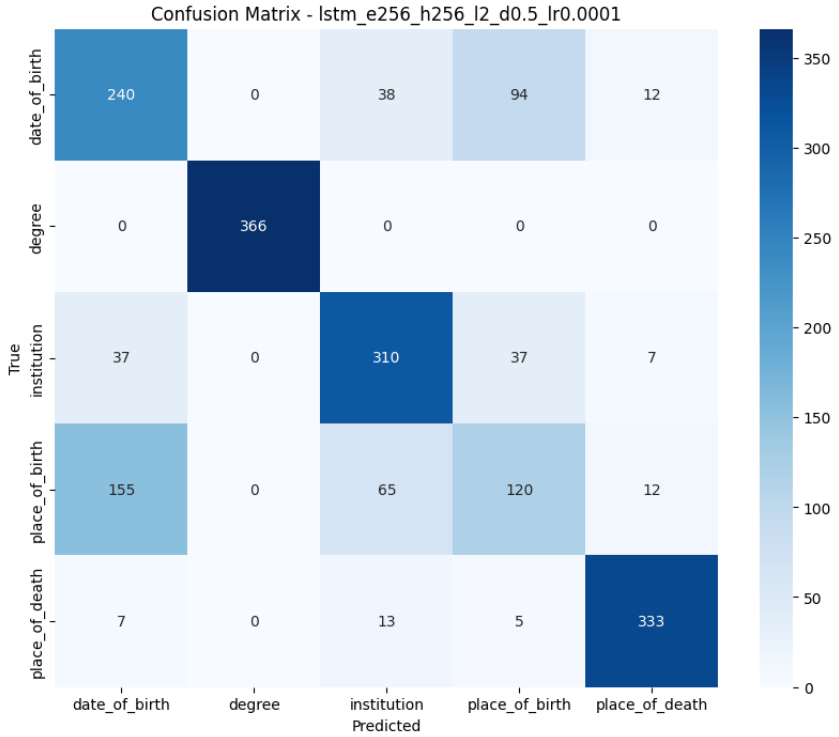

Figure 3: Confusion matrix for LSTM on the test set.

4.1.2 BiLSTM

- **Overall Performance**:
  - Accuracy: 81%
  - Macro F1-Score: 0.81
- **Strengths**:
  - Improved recall for challenging classes like `date_of_birth` and `place_of_birth`, thanks to its ability to capture bidirectional context.
- **Weaknesses**:
  - While it outperformed LSTM, it still lagged behind transformer-based models in overall accuracy and consistency across classes.

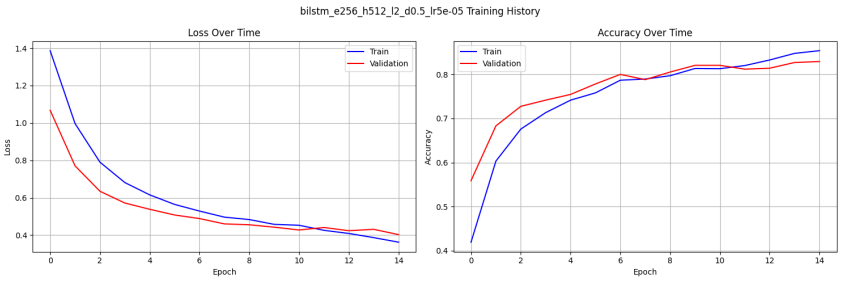

Figure 4: Training and validation loss and accuracy trends for BiLSTM.

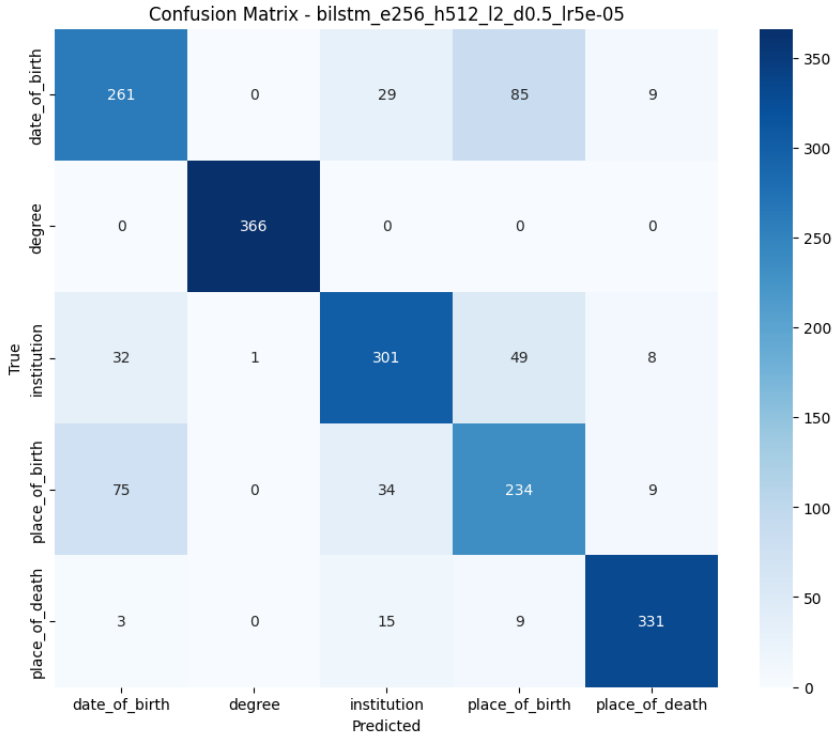

Figure 5: Confusion matrix for BiLSTM on the test set.

4.1.3 BERT

- **Overall Performance**:
  - Accuracy: 86%
  - Macro F1-Score: 0.86
- **Strengths**:
  - Delivered consistently high precision and recall across all relation types, including challenging ones like `place_of_birth`.
  - Its superior performance was attributed to its pretrained embeddings and powerful transformer architecture.
- **Weaknesses**:
  - Required more computational resources than sequence-based models, making it less suitable for resource-constrained setups.

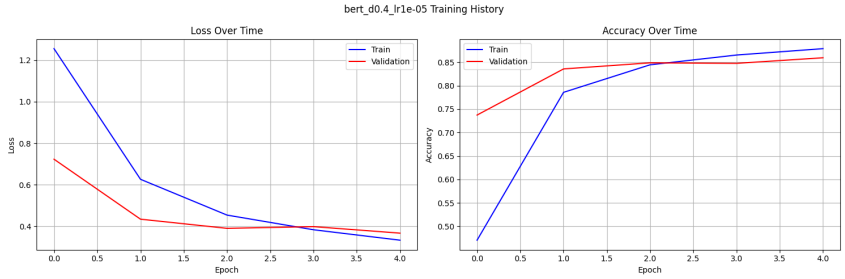

Figure 6: Training and validation loss and accuracy trends for BERT.

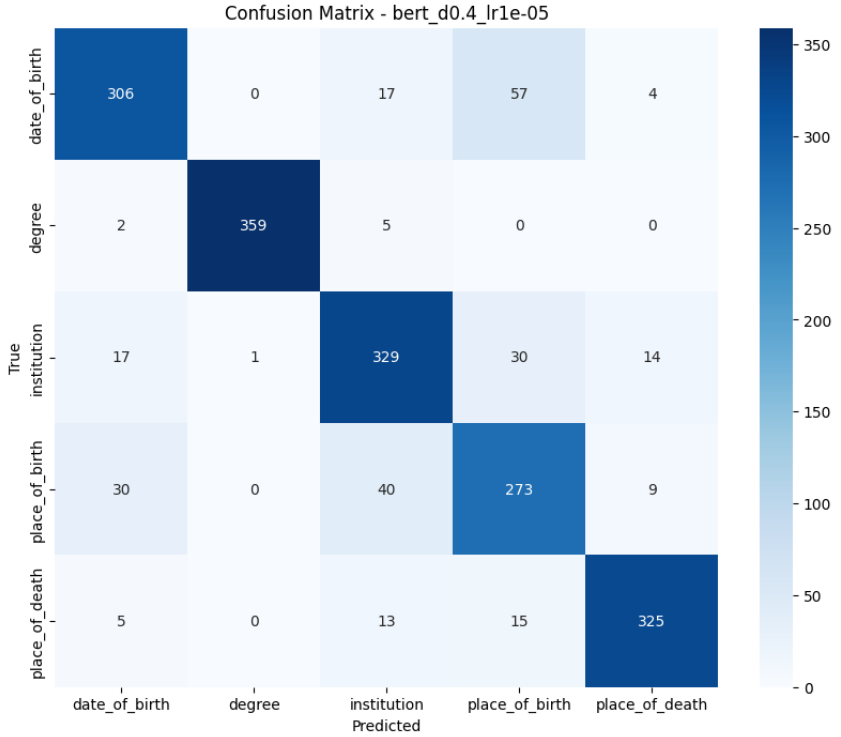

Figure 7: Confusion matrix for BERT on the test set.

4.1.4 RoBERTa

- **Overall Performance**:
  - Accuracy: 85%
  - Macro F1-Score: 0.85
- **Strengths**:
  - Showed strong performance on relations like `degree` and `place_of_death`, demonstrating its ability to handle contextual information effectively.
  - Its training stability, aided by a lower learning rate (3e-06), reduced the risk of overfitting.
- **Weaknesses**:

– Achieved slightly lower accuracy than BERT, potentially due to the higher dropout rate (0.5), which may have introduced too much regularization.

Figure 8: Training and validation loss and accuracy trends for RoBERTa.

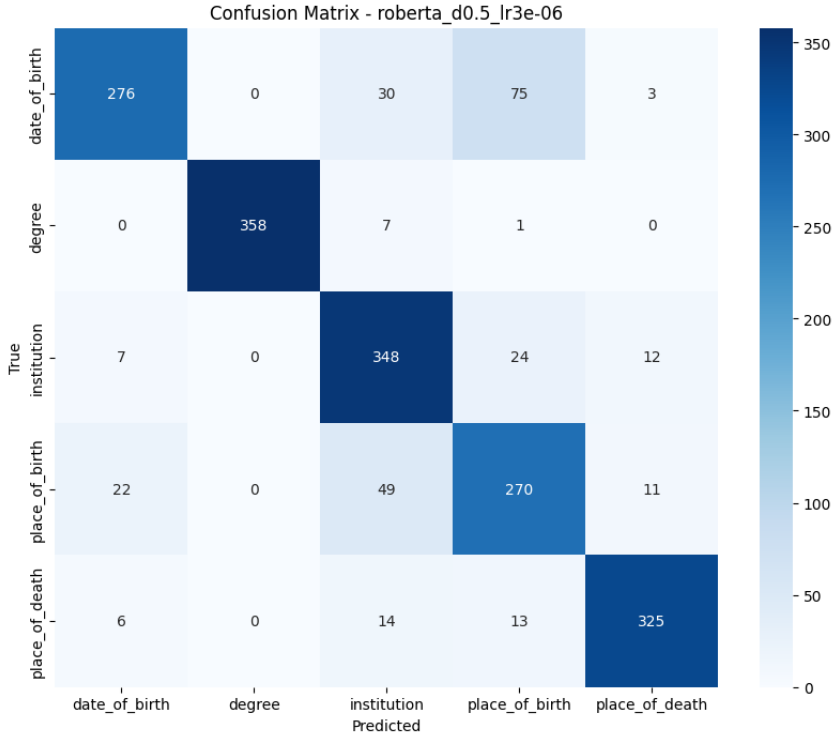

Figure 9: Confusion matrix for RoBERTa on the test set.

4.2 Knowledge Graph

The relationships extracted by the models were used to construct a knowledge graph (KG). The KG was analyzed using static and interactive visualizations to uncover patterns and insights.

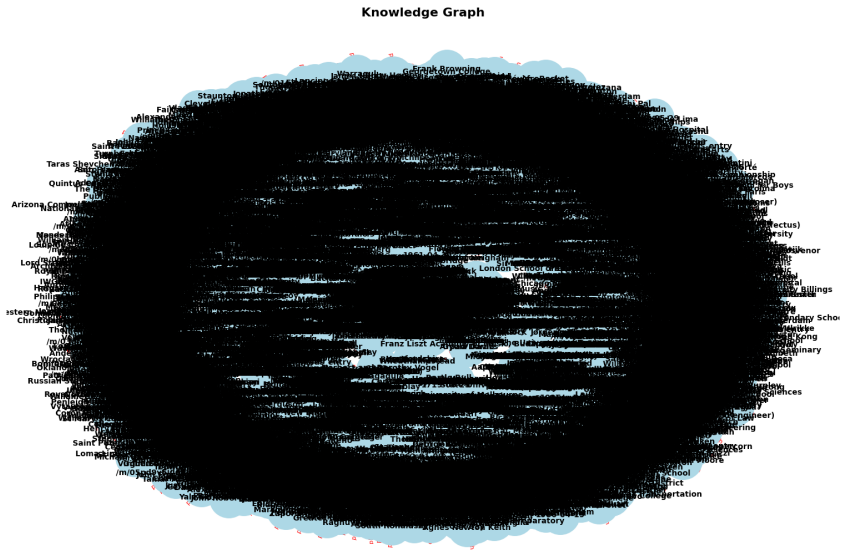

Figure 10: Static visualization of the knowledge graph.

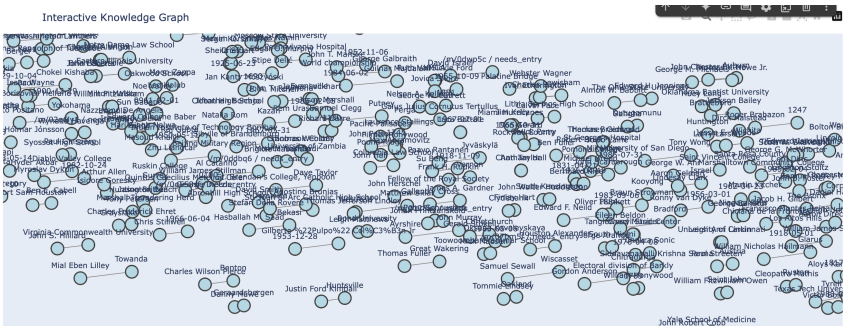

Figure 11: Interactive visualization of the entire knowledge graph.

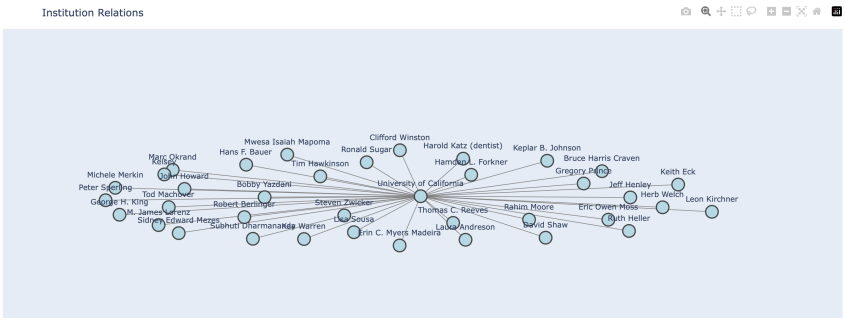

Figure 12: Interactive subgraph focusing on the `institution` relation.

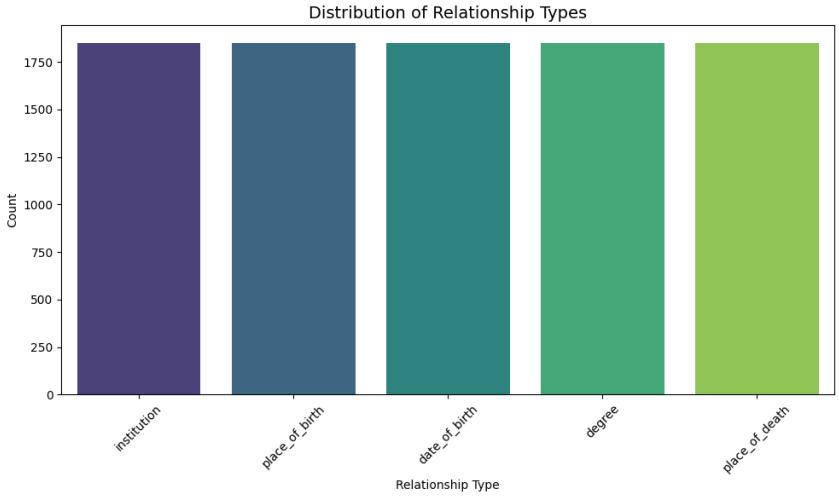

Figure 13: Distribution of relationship types in the dataset.

5 Discussion

5.1 Key Findings

The comparative analysis of LSTM, BiLSTM, BERT, and RoBERTa revealed several significant insights:

1. **BERT as the Top Performer**:
   - BERT achieved the highest accuracy (86%) and macro F1-score (0.86), demonstrating its superior ability to generalize across all relationship types.
   - Its success is attributed to its pretrained embeddings, bidirectional attention mechanism, and optimized hyperparameters, such as the learning rate (1e-05).
2. **RoBERTa's Close Performance to BERT**:
   - RoBERTa achieved a similar accuracy of 85% and macro F1-score of 0.85, proving to be a highly effective model.
   - Its lower learning rate (3e-06) contributed to its stability during training, helping it avoid overfitting.
   - However, its slightly higher dropout rate (0.5 compared to BERT's 0.4) may have introduced excessive regularization, resulting in slightly lower accuracy.
3. **BiLSTM as a Balanced Option**:
   - BiLSTM outperformed the standard LSTM, achieving an accuracy of 81%, by leveraging bidirectional context for better semantic understanding.
   - Its lower computational cost compared to transformer models makes it a practical choice for scenarios with limited resources.
4. **LSTM's Simplicity and Efficiency**:

- While LSTM lagged in accuracy (74%), it showed strong performance in simpler relations, such as `degree`.
- Its lightweight nature and faster training time make it an attractive option for smaller datasets or resource-constrained environments.

5.2 Challenges

Despite the promising results, the project encountered a few key challenges:

1. **Class Imbalance**:
   - Some relations, like `institution`, were heavily overrepresented in the dataset, while others, like `place_of_birth`, were underrepresented.
   - This imbalance affected model recall for underrepresented classes, particularly in sequence-based models like LSTM and BiLSTM.
2. **High Computational Costs**:
   - Transformer-based models like BERT and RoBERTa required significant computational resources and training time compared to LSTM and BiLSTM.
   - For example, BERT and RoBERTa took over three times longer to train than BiLSTM.
3. **Semantic Ambiguity in Relations**:
   - Some relationships, such as `place_of_birth` and `place_of_death`, were semantically similar, leading to occasional misclassifications across all models.
4. **Sensitivity to Hyperparameters**:
   - Incorporating class weights to address imbalance improved performance on underrepresented classes but required careful tuning, which added complexity to the training process.

5.3 Future Work

To address these challenges and improve outcomes, the following directions are proposed for future exploration:

1. **Advanced Fine-Tuning**:
   - Investigate more sophisticated fine-tuning techniques for transformer models, such as adapter layers or task-specific pretraining, to boost performance further.
2. **Integration of External Knowledge**:
   - Use external data sources (e.g., Wikipedia or Wikidata) to enrich the dataset and provide additional context for underrepresented relations.
3. **Scalability Enhancements**:

- Optimize training pipelines by incorporating distributed training or more efficient sampling techniques to handle larger datasets and reduce training time.

4. **Dynamic Class Weighting**:
   - Explore advanced methods for dynamically adjusting class weights during training to further address class imbalance issues.
5. **Evaluation of Lightweight Transformer Models**:
   - Experiment with more efficient transformer models like DistilBERT or ALBERT to balance performance and computational requirements.

6 Conclusion

This project successfully demonstrated the potential of integrating machine learning models into the process of building knowledge graphs (KGs). By comparing the performance of LSTM, BiLSTM, BERT, and RoBERTa, several meaningful insights were gained about their strengths, limitations, and practical trade-offs.

6.1 Key Takeaways

1. **Transformer Models Excel in Performance**:
   - BERT stood out as the best-performing model, achieving the highest accuracy (86%) and demonstrating consistent results across all relation types.
   - RoBERTa was a close second, with an accuracy of 85%. Its robustness during training and ability to handle complex relationships make it an excellent alternative to BERT.
2. **Sequence Models Offer Efficiency**:
   - LSTM and BiLSTM models, while less accurate than transformers, provided lightweight solutions with faster training times and lower resource requirements.
   - BiLSTM, in particular, struck a balance by achieving 81% accuracy while maintaining computational efficiency, making it a strong contender for environments with limited resources.
3. **Knowledge Graph Utility**:
   - The knowledge graph constructed during this project provided a clear visualization of the relationships between entities, showcasing the potential of automated relation extraction for building KGs at scale.
   - Subgraph analysis, like the focused exploration of the `institution` relation, revealed valuable patterns and connections between entities.

6.2 Practical Implications

This study highlights the feasibility of automating the KG construction process with advanced machine learning models. The trade-offs between computational cost and performance are now clearer, enabling better decision-making based on the specific requirements of different applications. Furthermore, the modular framework developed in this project can be extended and adapted for future improvements.

6.3 Final Thoughts

By bridging the gap between machine learning and knowledge representation, this project lays the groundwork for scalable and efficient KG construction. The insights gained here not only advance the automation of KGs but also provide a roadmap for future research. With its flexible architecture, the system can be enhanced to include additional relationships, larger datasets, and emerging machine learning models, paving the way for more sophisticated knowledge-driven applications.

7 References

1. Vaswani, A., et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

2. Devlin, J., et al. "BERT: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

3. Liu, Y., et al. "RoBERTa: A robustly optimized BERT pretraining approach." *arXiv preprint arXiv:1907.11692* (2019).

4. Hochreiter, S., Schmidhuber, J. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

5. Ji, S., et al. "A survey on knowledge graphs: Representation, acquisition, and applications." *IEEE Transactions on Neural Networks and Learning Systems* 33.2 (2021): 494-514.

6. Wang, Q., et al. "Knowledge graph embedding: A survey of approaches and applications." *IEEE Transactions on Knowledge and Data Engineering* 29.12 (2017): 2724-2743.

8 Dataset Link

Relation Extraction Corpus: https://github.com/google-research-datasets/relation-extraction-corpus