UNIVERSITÀ
DEGLI STUDI
DI MILANO

Language Models as Knowledge Base

By

Delnavaz Fotouhi

Prof. Alfio Ferrara

Information Retrieval

Department of Computer Science

Saturday 23rd March, 2024

Abstract

Relation prediction, the task of identifying and classifying relationships between entities in text, plays a crucial role in natural language processing applications such as information extraction, question answering, and knowledge graph construction. In this study, we present a BERT-based model for relation extraction, leveraging the power of contextual language representation to classify relations between entities. Our findings provide insights into the effectiveness of BERT-based models for relation prediction tasks and inform future research in the field.

Contents

List of Figures

Chapter 1

Introduction

Relation extraction is a fundamental task in natural language processing (NLP) that involves identifying and categorizing semantic relationships between entities mentioned in the text. This task is crucial for various NLP applications, including information retrieval, question answering, and knowledge graph construction. Traditional approaches to relation extraction often rely on handcrafted features and domain-specific rules, which are limited in their ability to capture the rich contextual information present in natural language.

Recently, transformer-based models, such as BERT (Bidirectional Encoder Representations from Transformers), have emerged as powerful tools for NLP tasks, thanks to their ability to capture contextual information and learn representations directly from raw text. In this study, we explore the application of BERT for relation extraction, aiming to leverage its contextual embeddings to classify relations between entities accurately.

Our study investigates the effectiveness of a BERT-based model for relation prediction tasks, focusing on tasks such as identifying relationships between people, locations, dates, and educational degrees mentioned in text. We evaluate the model's performance on a benchmark dataset and analyze its ability to handle challenges such as class imbalance.

Through our analysis, we aim to provide insights into the capabilities and limitations

of BERT-based models for relation extraction tasks and contribute to the ongoing research efforts in the field of NLP and information extraction.

Chapter 2

Data Manipulation

In the initial phase of our project, we began by manipulating the dataset to prepare it for subsequent processing. This phase involved several crucial steps.

2.1 Attribute Analysis and Visualization

We began by examining the attributes present in our dataset and understanding the distribution of relation types (predicates). There are five types of relation: institution, place of birth, date of birth, place of death, and degree. Each predicate identifies a relation between two entities naming subject and object. Figure 2.1 depicts the distribution of data over these relation types.

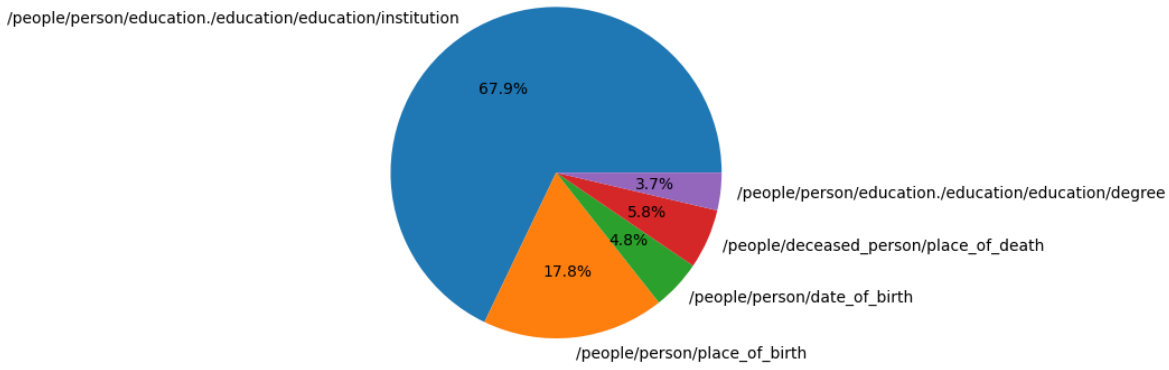

Figure 2.1: Labels Distribution

2.2 Text Length Analysis and Filtering

Understanding the distribution of text lengths in the dataset is essential for effective model training. We conducted an analysis of text lengths to examine the distribution of data regarding text lengths. The process involved the following steps.

- Calculating the frequency distribution of text lengths.
- Visualizing the distribution using a bar plot.
- Filtering out data points with text lengths exceeding a predefined threshold.

The plot shows that the majority of texts have a length of less than 1000 characters. Therefore, for the sake of simplicity and efficiency, we filter out the text with a length longer than 600 characters.

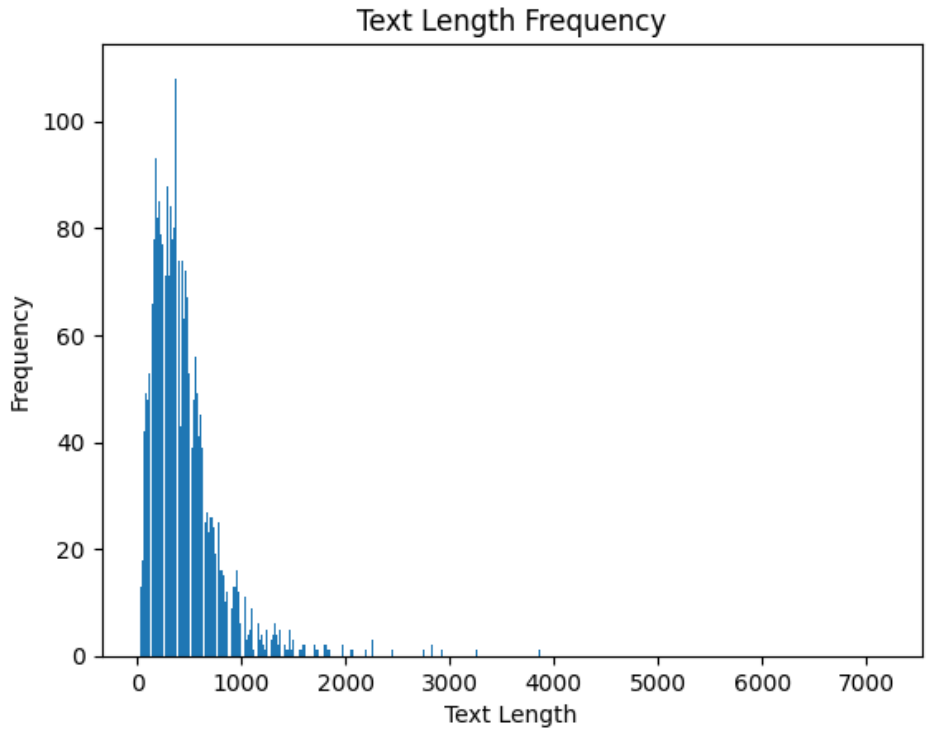

Figure 2.2: Text Length Frequency

2.3 Data Balancing

As illustrated in Figure 2.1, the dataset exhibits significant class imbalance, with the 'institution' class representing approximately 70 percent of the data. This imbalance can cause some problems in the training of the model. To address class imbalance issues within the dataset, we implemented both undersampling and oversampling techniques This step involved:

- undersampling the majority classes('institution', 'place of birth', and 'place of death')
- oversampling the minority classes('date of birth', and 'degree')

The result is a data set with 11000 data points containing five classes, and each class contains 2200 data points.

2.4 Entity Marking in Text

In preparation for subsequent processing steps, we marked entity mentions within the text, naming subject and object. This task involved identifying subject and object entities within each text snippet and marking their occurrences with special tokens for further analysis and processing. The tokens denoting the start and end of a subject are [E1] and [/E1] respectively. Likewise, [E2] and [/E2] denote the start and end of objects.

2.5 Label Encoding

To facilitate model training, we encoded the relation labels into numerical format. This step involved mapping relation types to numerical labels using a label encoder. The encoded labels were then incorporated into the dataset for training purposes.

Chapter 3

Data preparation

After the initial data manipulation phase, we proceeded with the data preprocessing stage, which involved tokenizing and preparing the text data for ingestion into the BERT model. The key steps undertaken during this phase are outlined below.

3.1 Tokenization with BERT

We employed the BERT tokenizer to preprocess raw text data, transforming it into tokenized sequences suitable for model input. Utilizing the bert-base-uncased pretrained model, the tokenizer processed each input sequence, tokenizing it into constituent tokens. To adhere to the maximum sequence length requirements of BERT, which in our case was set to 128 tokens, we truncated or padded the tokenized sequences accordingly. Additionally, we augmented each sequence with special tokens, [CLS] and [SEP], denoting the beginning and end of the sequence, respectively. Subsequently, we converted these tokenized sequences into input IDs by assigning a unique numeric value to each token. These input IDs form the numerical representation of the tokenized text, serving as the input data fed into the model for further processing.

3.2 Data Splitting

The dataset was split into training, validation, and test sets to facilitate model training, hyperparameter tuning, and evaluation. The training set comprised 70 percent of the total data, while the validation set and the test set each accounted for 15 percent of the data. The "train-test-split" function from scikit-learn was employed for this purpose. Additionally, to ensure a balanced distribution of relation types across the splits, we utilized the *'stratify'* parameter during splitting.

3.3 DataLoader Instantiation

We created TensorDataset instances from the input tensors and labels, thereby enabling seamless integration with PyTorch's data handling utilities. Subsequently, DataLoader instances were instantiated for the training, validation, and test sets, enabling efficient batch processing during model training and evaluation.

Chapter 4

Model Training

In this chapter, we trained and evaluated our BERT-based model for relation prediction. The process involved several key steps, including model initialization, training, and evaluation of model performance. Below is an overview of each step:

4.1 Model Architecture

Our model architecture is based on BERT (Bidirectional Encoder Representations from Transformers), a state-of-the-art language representation model developed by Google. BERT utilizes a transformer architecture, which enables it to capture contextual information from both the left and right contexts of a word in a sentence. This bidirectional context encoding is particularly beneficial for tasks like relation extraction, where understanding the surrounding context is crucial for accurate classification. In our implementation, we used the bert-base-uncased variant of BERT, which consists of 12 transformer layers and 768 hidden units. We added a classification layer on top of the pre-trained BERT model to perform relation classification. This classification layer maps the encoded input representation to the appropriate relation label using a softmax activation function. The number of labels was set to 5 to accommodate the five relation types present in our dataset.

4.2 Model Training

The model was trained using the AdamW optimizer with a learning rate of 2e-5. We utilized mixed precision training to improve training efficiency and speed up convergence. The training loop iterated over multiple epochs, with the model's performance monitored on the validation set at regular intervals.

4.3 Model Evaluation

After training, we evaluated the model's performance on the validation and test sets. We computed metrics such as accuracy, precision, recall, and F1-score to assess the model's classification performance. Additionally, we generated a confusion matrix to visualize the model's predictions and identify potential areas of improvement.

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0	0.47	0.28	0.35	368
1	0.75	0.58	0.66	318
2	0.35	0.22	0.27	307
3	0.37	0.52	0.43	347
4	0.44	0.72	0.55	310
Accuracy	-	-	0.46	1650
Macro Avg	0.48	0.47	0.45	1650
Weighted Avg	0.48	0.46	0.45	1650

Table 4.1: Classification Report

Table 4.1 highlights the model's performance across different classes, indicating satisfactory results for "place of death", "date of birth", and "degree". However, noticeable room for enhancement exists particularly in the classification of 'place of birth' and 'institution' classes.

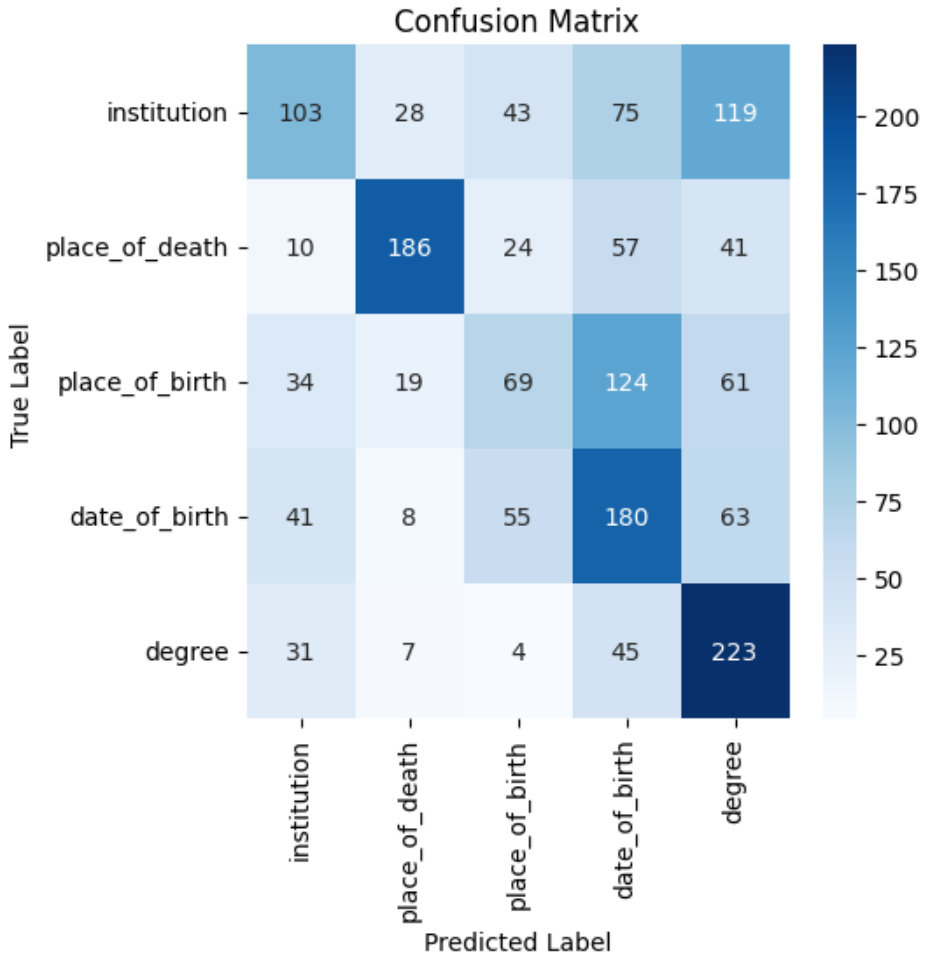

Figure 4.1: Caption

Chapter 5

Conclusion

In this study, we developed and evaluated a BERT-based model for relation extraction. Our model leveraged the powerful contextual representations learned by BERT to classify relations between entities in text. Through a comprehensive training and evaluation process, we gained valuable insights into the model's performance and its ability to generalize to unseen data.

The results of our evaluation revealed both strengths and areas for improvement in the model's performance. While achieving an overall accuracy of 46 percent, the model exhibited varying levels of precision, recall, and F1-score across different relation types. Classes such as 'place of death' and 'degree' demonstrated relatively higher precision, recall, and F1-score, indicating the model's proficiency in identifying these relations. Conversely, classes such as 'place of birth' and 'institution' displayed lower performance metrics, highlighting potential challenges in accurately classifying these relations.

In conclusion, our study provides valuable insights into the effectiveness of BERT-based models for relation extraction tasks. By understanding the model's strengths and weaknesses, we can inform future research and development efforts aimed at enhancing its performance and applicability in real-world applications.