Relation Extraction on TACRED Dataset

Bader Shalata, Dmitrii Naumov, Minh Duc Pham, Kuan-Ling Liu
University of Manchester

Abstract

Relation extraction (RE) identifies and categorizes relationships between named entities in text. This study evaluates two RE approaches using the TACRED dataset: 1) a deep learning approach without transformers, Long Short-Term Memory (LSTM), and 2) a deep learning approach based on transformer models, including BERT and GPT. We also categorize sentence length in our evaluation. Models are assessed using precision, recall, and F1-score, highlighting their strengths, weaknesses, and providing insights into effective relation extraction techniques.

1 Introduction

Relation extraction (RE) identifies semantic links in unstructured text, playing a crucial role in information retrieval, knowledge base construction, and question-answering systems. A key dataset for RE is TACRED (TAC Relation Extraction Dataset), which contains 106,264 instances across 42 relation types. It focuses on extracting relationships between named entities like persons, organizations, and locations. However, TACRED presents several challenges, including complex sentence structures, diverse linguistic patterns, and severe class imbalance, with 80% of instances labeled as 'no_relation.' These factors make it difficult for models to capture meaningful relationships and generalize effectively, posing significant obstacles for classification models. We evaluated two RE techniques: a deep learning-based approach without transformer, LSTM; and deep learning with transformer models, BERT and GPT. The paper is structured as follows: we first review related works, then explain the methodology and models. Next, we present the findings, followed by a discussion and future directions.

Relation	Count
`no_relation`	84,491
`per:title`	3,864
`org:top_members/employees`	2,770
`per:employee_of`	2,163
`org:alternate_names`	1,359

Table 1: Top 5 most frequent relation types

2 Related Works

Relation extraction (RE) has evolved from traditional sequence-based models to deep learning techniques. A major advancement came with TACRED (Zhang et al., 2017), a large-scale dataset designed to improve supervised RE. TACRED introduced new challenges, particularly longer sentence structures, diverse linguistic patterns, and extreme class imbalance, where the majority of instances belong to the 'no_relation' category. These challenges made relation classification difficult for traditional models, requiring more sophisticated architectures.To address these challenges, Zhang et al. (2017) proposed a position-aware attention-based LSTM model, achieving a 65.1% F1-score by leveraging entity position embeddings. While effective, LSTMs still struggled with capturing long-range dependencies, limiting their ability to model complex relationships.This limitation led to the adoption of transformer-based architectures like BERT (Devlin et al., 2019), which leverages contextualized embeddings and self-attention mechanisms to process entire sequences in parallel, improving RE performance. BERT-based models have since outperformed LSTMs on TACRED, addressing challenges related to sentence complexity and entity interactions (Stoica et al., 2021).

3 Methodology

3.1 Bi-LSTM

The bidirectional LSTM model can provide several advantages for relation extraction. The based method of this model that uses only word embedding could achieve good results and adding more features can even improve the overall performance (Zhang et al., 2015). Its bidirectional aspect efficiently captures contextual information on both sides of entity pairs, which is crucial because the relation indicators can be anywhere in a sentence. Furthermore, the model's sequential processing feature is well-aligned with natural language structure, which enables it to identify patterns in word sequences that can demonstrate particular relationships. Our extraction model uses a Bidirectional Long-Short-Term Memory neural network to identify the relationships between entities. The model handles input via multiple embedding layers, such as work embeddings to capture lexical data, and position embeddings capturing distances between tokens and target entities. Meanwhile, named-entity recognition labels and part-of-speech tags provide semantic and syntactic information. Categorical information on target entities was also combined into the representation via entity-type embedding. Such features were then concatenated and fed into the bidirectional LSTM layer, which processed the sequences in both directions to capture contextual information throughout the whole sentence. Next, the output of Bidirectional LSTM was passed through a fully connected layer with batch normalization. Such output results were produced in the form of raw logits, which then would be used to decide on the prediction label.

Data Preprocessing: A data processing pipeline was implemented for the BiLSTM-based relation extraction model to transform relevant information from the data set into a structured format suitable for training the model. Vocabulary construction was conducted based on certain information within the dataset such as tokens, entity types, part-of-speech tags, and named entity recognition labels. Textual elements can be mapped to indices for the neural network to process and study using such vocabularies. Next,the feature engineering process transformed each observation in the training dataset into a multidimensional form. Meanwhile, spatial relationships within the text were captured by calculating positional features based on relative distances between each token and the entities' centers. Thus, entity representation was also converted to its indices with respect to its entity type vocabulary. All features including word tokens, part-of-speech tags, named entity recognition tags, entity type indices, and positional features were consolidated into instances with their corresponding labels encoded in numerical form. Moreover, to enhance processing efficiency for variable-length sequences with packed sequence operations, a custom collate function along with length-based sorting was used fo loading the dataset. Furthermore, to optimize processing efficiency for variable-length sequences through packed sequence operations, a custom collate function with length-based sorting was created and utilized while loading the datasets. This approach could help process similar-length sequences together when executing batch processing despite sequence length variation. These preprocessing steps were consistently applied during the inference stage to maintain data transformation integrity, ensuring the model's capacity to generalize with new samples.

Fine Tuning: Other techniques were adopted with the aim of improving the performance of the model.A dropout rate of 0.3 for regularization was defined to prevent overfitting by deactivating 30% of the neurons during the training process. To tackle the imbalance aspect of the dataset where the 'no_relation' class dominates, the under-sampling technique was utilized to reduce the majority class samples while maintaining minority class instances, providing equitable data distribution (Spelmen and Porkodi, 2018). A maximum ratio parameter was set to 3 as the proportion between dominant and minority classes. After balancing, the imbalance ratio between 'no_relation' and other labels was reduced from $4.2:1$ to $3:1$. Thus, class-weighted loss function was also used to prioritize minority labels during training. Regarding the optimizer, AdamW was chosen with a weight decay of 1e-5 for regularization. Meanwhile, learning rate was reduced automatically when validation metrics improved. Last but not least, early stopping with 5-epoch patience was implemented to prevent overfitting and conserve compute resources when F1 score stopped improving.

3.2 BERT

For this approach, the pre-trained BERT base uncased, a transformer encoder was utilized, this

model was initially trained on BookCorpus which consisted of 11,038 unpublished books and English Wikipedia (Devlin et al., 2019).

BERT base consists of 12 transformer layers that each contain multi-head self-attention mechanisms with feed-forward networks. Each of these transformer layers operates on hidden representations of size 768, meaning each token is represented by a vector of 768 dimensions. Each multi-head self-attention mechanism consists of 12 attention heads, allowing the model to capture diverse types of relationships between tokens at each layer. The *attention layers* play a crucial role by computing attention scores between all token pairs, which allows each token to attend to all other tokens in the sentence. This helps the model dynamically focus on the most relevant parts of the input when constructing contextual embeddings for each token. This *architecture* utilizes a bidirectional self-attention mechanism which allows the model to capture context from both the left and right sides of a specific token in a sentence.As a result, each token is represented with a contextual embedding rather than a static one unlike traditional word embeddings, making this approach particularly effective for our relation extraction task as these contextual embeddings improve the model's understanding of entity and token relationships.

Fine Tuning: Although BERT is proficient in understanding word relationships, it was not originally designed for relation extraction tasks. To adapt BERT to this application, the model architecture was modified by adding an additional transformer encoder layer with 12 attention heads, enhancing contextual representations and improving the model's attention mechanism, particularly for long-range dependencies. Furthermore, a dropout rate of 0.2 was applied before feeding the embeddings into the classification head to prevent overfitting. The classification head consisted of one hidden layer with ReLU activation, introducing non-linearity, and a final output layer containing 42 neurons (corresponding to the number of classes), which are responsible for predicting entity relationships.

Data Pre-processing: The initial dataset included features like Part-of-Speech tags and dependency parsing, which were not needed for the BERT model. BERT learns contextual representations through self-attention, capturing word dependencies. Therefore, only sentence tokens, subject (from subj start and subj end positions), and object (from obj start and obj end positions) were used. The data was then tokenized using the BERT base uncased tokenizer, creating tensors with input IDs, attention masks (for padding), labels, and entity indices. These tensors were stored in TensorDatasets and loaded in batches of 50, chosen to ensure diverse class labels in each batch.

Implementation: For the training, we have trained the model for 10 epochs, with AdamW optimizer and learning rate of 5×10^{-5} with a weight decay of 0.01, which introduces regularization into the loss function, preventing overfitting problems. Furthermore, we implemented a scheduler to dynamically adjust the learning rate during training with step size of 500 and gamma of 0.9 meaning the learning rate is reduced by 10% every 500 batches. For the loss function, we selected a custom version of CrossEntropyLoss that addresses class imbalance by applying an inverse weighting scheme, which gives more weight and penalty to minority class. The weight for each class i is this:

$$W_i = \frac{1}{\sqrt{N_i}}$$

Where:
Wi is the weight for class *i*,
Ni is the number of samples for class *i*

4 Results and Discussions

To perform performance evaluation of the trained models on a highly imbalanced dataset, we have decided to keep track of macro averages of precision, recall and f1 score metrics. While weighted average is expected to provide a performance estimate for a degree of our models' fit to the testing data, macro-average would provide insights into models' performance under the assumption of equal importance of all featured relations. All deep learning models were trained on a single-stage classification task in which detection of the existence of a relationship was represented by an additional relation class during training. Whereas LSTM model stroke the balance between macro and micro accuracy metrics, both transformer models were prone to excessively focus on the largest no-relation class despite an applied up-sampling and class-occurrence-based loss reweighting. In fact GPT2 has failed training by stalling in a local minimum and leaning fully on the largest class in the dataset. Overall, at this stage, no model achieved sufficiently adequate performance levels for an accurate and reliable relation extraction

functionality, considering equal importance of relations. However, LSTM and BERT models were still able to perform well at a limited number of classes - most commonly at classes that were better represented in the dataset.

	Weighted			Macro		
	LSTM	**BERT**	**GPT2**	**LSTM**	**BERT**	**GPT2**
P	0.81	0.82	0.73	0.36	0.41	0.26
R	0.79	0.80	0.79	0.43	0.45	0.18
F1	0.79	0.81	0.74	0.36	0.41	0.18

Table 2: Performances on no_relation and relation data

Since the general relation classification task can be naturally divided into two sequential classification stages of relation detection and its definition, we have decided to train all models exclusively on the subset of the data that were labeled as having any relationship between entities in sentences. This approach builds on the rationale that isolating learning of such disparate learning objectives can increase smoothness of training and enable transformer models to better leverage semantic assessment capabilities. And indeed, the reduced training goal led to dramatic boost across all monitored metrics. F1 score increased by values ranging from 0.15 to almost 0.3. LSTM was a model that particularly benefited from loosening the classification task formulation, allowing it to almost double its f1 score, bringing it to the performance level that is slightly superior to BERT. It might happen due to inclusion of auxiliary linguistic features, which is contrasted with transformer-based models, that operate exclusively on hidden representations of contextualized tokens. Relation detection part can be quite efficiently performed even by traditional ML approaches, like SVM, leveraging such linguistic features as POS tags, dependency structures and entity labels. We managed to train SVM model that reached an f1 score of about 0.9, meaning that this stage is able to pass entities for a further classification with a reasonably small error. It also suggests, that keeping a learning of a 'no-relation' class separated in its parameters from a general classification of a relationship nature significantly improves RE for every model architectures. Additionally, this model comparison illustrates the relative advantage of having bi-directional contextual awareness for RE task. Decoder only GPT is at least 0.2 points below the models that are allowed to attend to future tokens in a sentence (bi-LSTM, BERT) in terms of accuracy metrics in both training setups.

	Weighted			Macro		
	LSTM	**BERT**	**GPT2**	**LSTM**	**BERT**	**GPT2**
P	0.82	0.71	0.60	0.66	0.6	0.41
R	0.83	0.70	0.55	0.61	0.58	0.45
F1	0.82	0.69	0.55	0.61	0.57	0.43

Table 3: Performances on relation data

Despite possible varying complexity of RE, depending on the textual sequence length, f1 appears to be fairly consistent for the all sentence categories based on a number of tokens they comprised of. The only model that experiences a decline in performance on longer sequences is BERT, which has its f1 score decreased withing the top 20% sentences by its length.

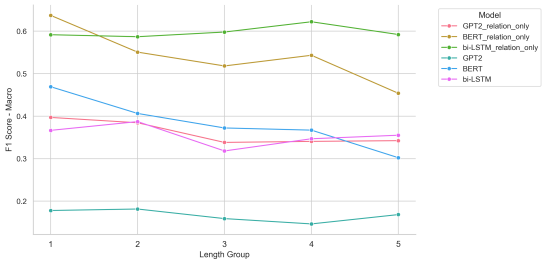

Figure 1: Macro F1 Score Across Groups of Sentences with Varying Sequence Length (Groups represent cohorts containing 20% of the whole dataset and are positioned in an ascending order of a number of tokens in sentences (From shortest to longest sentences)

5 Conclusion

Even after decoupling training for relation detection and classification stages, maximum accuracies achieved on a test set remains at the moderate levels, making it ill-suited for a fully automated relation extraction pipelines. The distribution of classes' accuracy, precision and recall scores appears to be highly uneven, simultaneously allowing for almost perfectly predicted relations and classes almost fully overlooked by models. Partially the lack of capacity to improve stems from severe imbalances in classes, that cannot be fully rectified by up-sampling. Nevertheless lower metrics does not come from overfitting, implying that primitiveness of classification can act as a constraint for model to improve separability of classes. Potential refinement of a classification component and its transformation from a single fully-connected module into multiple heads (channels) based on groups of features (subject and object special tokens, broader sentence context using mean and max-pooling), along with implementation of an entity-aware attention, might have a capacity to even out accuracy metrics for all classes present in the dataset.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Vimalraj S Spelmen and R Porkodi. 2018. A review on handling imbalanced data. In *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, pages 1–11.

George Stoica, Emmanouil Antonios Platanios, and Barnab'as P'oczos. 2021. Re-tacred: Addressing shortcomings of the tacred dataset. In *AAAI Conference on Artificial Intelligence*.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 73–78, Shanghai, China.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

A Appendix

A.1 SVM

We also implemented a traditional machine learning approach using Support Vector Machines (SVM), but it is not included in our main discussion. The method involved a binary SVM classifier to distinguish 'no_relation' from relational data, followed by a multi-class SVM to classify specific relation types.

	Weighted SVM	**Macro SVM**
P	0.71	0.71
R	0.62	0.62
F1	0.58	0.58

Table 4: Macro average of SVM include no relation and relation data

	Weighted SVM	**Macro SVM**
P	0.79	0.63
R	0.78	0.62
F1	0.77	0.61

Table 5: Macro average of SVM include only relation data

Class	**Precision**	**Recall**	**F1-score**	**Support**
no_relation	0.57	0.94	0.71	3325
relation	0.84	0.30	0.44	3325
accuracy			0.62	6650
macro avg	0.71	0.62	0.58	6650
weighted avg	0.71	0.62	0.58	6650

Table 6: SVM Binary Classification

Class	Precision	Recall	F1-score	Support
org:alternate_names	0.86	0.79	0.82	213
org:city_of_headquarters	0.99	0.94	0.96	82
org:country_of_headquarters	0.91	0.97	0.94	108
org:dissolved	0.50	0.50	0.50	2
org:founded	0.97	0.97	0.97	37
org:founded_by	0.55	0.09	0.15	68
org:member_of	0.12	0.17	0.14	18
org:members	0.00	0.00	0.00	31
org:number_of_employees/members	1.00	1.00	1.00	19
org:parents	0.35	0.37	0.36	62
org:political/religious_affiliation	1.00	1.00	1.00	10
org:shareholders	0.00	0.00	0.00	13
org:stateorprovince_of_headquarters	1.00	0.96	0.98	51
org:subsidiaries	0.21	0.36	0.27	44
org:top_members/employees	0.69	0.73	0.71	346
org:website	1.00	1.00	1.00	26
per:age	1.00	1.00	1.00	200
per:alternate_names	0.14	0.55	0.23	11
per:cause_of_death	1.00	1.00	1.00	52
per:charges	1.00	1.00	1.00	103
per:children	0.36	0.35	0.36	37
per:cities_of_residence	0.84	0.95	0.89	189
per:city_of_birth	0.33	1.00	0.50	5
per:city_of_death	0.50	0.07	0.12	28
per:countries_of_residence	0.78	0.91	0.84	148
per:country_of_birth	0.00	0.00	0.00	5
per:country_of_death	0.00	0.00	0.00	9
per:date_of_birth	0.62	0.56	0.59	9
per:date_of_death	0.93	0.94	0.94	54
per:employee_of	0.58	0.69	0.63	264
per:origin	0.88	0.81	0.85	132
per:other_family	0.37	0.62	0.46	60
per:parents	0.56	0.40	0.47	88
per:religion	1.00	1.00	1.00	47
per:schools_attended	1.00	0.57	0.72	30
per:siblings	0.95	0.35	0.51	55
per:spouse	0.79	0.68	0.73	66
per:stateorprovince_of_birth	0.43	0.38	0.40	8
per:stateorprovince_of_death	0.00	0.00	0.00	14
per:stateorprovinces_of_residence	0.80	0.83	0.81	81
per:title	1.00	1.00	1.00	500
accuracy			0.78	3325
macro avg	0.63	0.62	0.61	3325
weighted avg	0.79	0.78	0.77	3325

Table 7: SVM Multi-class Classification Results

Class	Precision	Recall	F1-score	Support
no_relation	0.90	0.84	0.87	12184
org:alternate_names	0.62	0.68	0.65	213
org:city_of_headquarters	0.46	0.71	0.56	82
org:country_of_headquarters	0.40	0.60	0.48	108
org:dissolved	0.00	0.00	0.00	2
org:founded	0.60	0.70	0.65	37
org:founded_by	0.32	0.10	0.16	68
org:member_of	0.00	0.00	0.00	18
org:members	0.00	0.00	0.00	31
org:number_of_employees/members	0.29	0.37	0.33	19
org:parents	0.04	0.02	0.02	62
org:political/religious_affiliation	0.17	0.70	0.27	10
org:shareholders	0.00	0.00	0.00	13
org:stateorprovince_of_headquarters	0.46	0.76	0.58	51
org:subsidiaries	0.33	0.14	0.19	44
org:top_members/employees	0.57	0.79	0.66	346
org:website	0.45	0.92	0.61	26
per:age	0.62	0.85	0.72	200
per:alternate_names	0.00	0.00	0.00	11
per:cause_of_death	0.57	0.31	0.40	52
per:charges	0.56	0.87	0.68	103
per:children	0.29	0.14	0.19	37
per:cities_of_residence	0.35	0.52	0.42	189
per:city_of_birth	0.25	0.40	0.31	5
per:city_of_death	0.46	0.21	0.29	28
per:countries_of_residence	0.33	0.48	0.39	148
per:country_of_birth	0.00	0.00	0.00	5
per:country_of_death	0.00	0.00	0.00	9
per:date_of_birth	0.50	0.78	0.61	9
per:date_of_death	0.41	0.22	0.29	54
per:employee_of	0.42	0.62	0.50	264
per:origin	0.42	0.58	0.49	132
per:other_family	0.00	0.00	0.00	60
per:parents	0.51	0.23	0.31	88
per:religion	0.41	0.85	0.56	47
per:schools_attended	0.41	0.47	0.44	30
per:siblings	0.48	0.51	0.50	55
per:spouse	0.37	0.39	0.38	66
per:stateorprovince_of_birth	0.24	0.50	0.32	8
per:stateorprovince_of_death	0.67	0.14	0.24	14
per:stateorprovinces_of_residence	0.45	0.62	0.52	81
per:title	0.60	0.89	0.71	500
accuracy			0.79	15509
macro avg	0.36	0.43	0.36	15509
weighted avg	0.81	0.79	0.79	15509

Table 8: BiLSTM Classification Results on all dataset

A.2 Bi-LSTM

Bi-LSTM model trained on the full dataset

- **Train Resluts**:
  - **Train Loss**: 0.0165
  - **Dev Loss**: 4.7232
  - **Accuracy**: 0.7718
  - **Precision**: 0.4285
  - **Recall**: 0.4735
  - **F1**: 0.4225
- **Test Results**:
  - **Accuracy**: 0.7899
  - **Precision**: 0.3559
  - **Recall**: 0.4266
  - **F1**: 0.3637

Bi-LSTM model trained without no_relation observations

- **Train Resluts**:
  - **Train Loss**: 0.0064
  - **Dev Loss**: 2.4283
  - **Accuracy**: 0.8138
  - **Precision**: 0.6877
  - **Recall**: 0.6557
  - **F1**: 0.6543
- **Test Results**:
  - **Accuracy**: 0.8298
  - **Precision**: 0.6634
  - **Recall**: 0.6103
  - **F1**: 0.6119

Class	Precision	Recall	F1-score	Support
org:alternate_names	0.82	0.84	0.83	213
org:city_of_headquarters	0.95	0.99	0.97	82
org:country_of_headquarters	0.91	0.97	0.94	108
org:dissolved	0.00	0.00	0.00	2
org:founded	0.94	0.92	0.93	37
org:founded_by	0.73	0.16	0.27	68
org:member_of	0.29	0.11	0.16	18
org:members	0.20	0.03	0.06	31
org:number_of_employees/members	1.00	1.00	1.00	19
org:parents	0.29	0.39	0.33	62
org:political/religious_affiliation	1.00	1.00	1.00	10
org:shareholders	1.00	0.08	0.14	13
org:stateorprovince_of_headquarters	1.00	0.98	0.99	51
org:subsidiaries	0.34	0.41	0.37	44
org:top_members/employees	0.84	0.99	0.91	346
org:website	1.00	1.00	1.00	26
per:age	1.00	1.00	1.00	200
per:alternate_names	0.10	0.18	0.13	11
per:cause_of_death	1.00	1.00	1.00	52
per:charges	1.00	1.00	1.00	103
per:children	0.26	0.49	0.34	37
per:cities_of_residence	0.85	0.97	0.91	189
per:city_of_birth	0.50	0.40	0.44	5
per:city_of_death	0.69	0.32	0.44	28
per:countries_of_residence	0.77	0.91	0.83	148
per:country_of_birth	0.00	0.00	0.00	5
per:country_of_death	0.00	0.00	0.00	9
per:date_of_birth	1.00	0.67	0.80	9
per:date_of_death	0.95	1.00	0.97	54
per:employee_of	0.95	0.96	0.95	264
per:origin	0.88	0.80	0.84	132
per:other_family	0.30	0.27	0.28	60
per:parents	0.47	0.19	0.27	88
per:religion	1.00	1.00	1.00	47
per:schools_attended	0.64	0.53	0.58	30
per:siblings	0.49	0.49	0.49	55
per:spouse	0.48	0.61	0.53	66
per:stateorprovince_of_birth	0.40	0.50	0.44	8
per:stateorprovince_of_death	0.33	0.07	0.12	14
per:stateorprovinces_of_residence	0.81	0.80	0.81	81
per:title	1.00	1.00	1.00	500
accuracy			0.83	3325
macro avg	0.66	0.61	0.61	3325
weighted avg	0.82	0.83	0.82	3325

Table 9: BiLSTM Classification Results without no_relation data

A.3 BERT

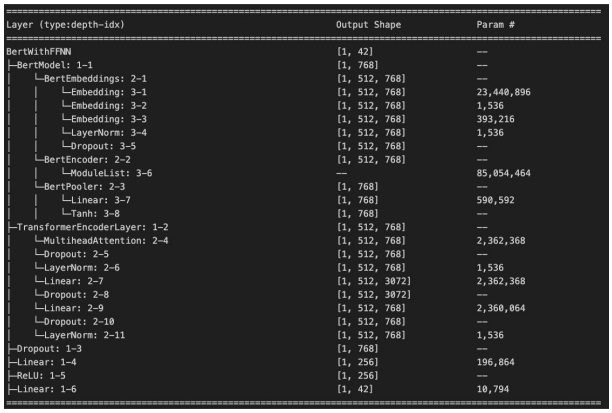

Figure 2: BERT Architecture after Fine Tuning

BERT Overall Test Set Metrics on all data

- **Precision**: 0.8148
- **Recall**: 0.8025
- **F1-score**: 0.8061

Class	Precision	Recall	F1-Score	Support
0	0.41	0.28	0.33	47
1	0.30	0.30	0.30	81
2	0.25	0.14	0.18	14
3	0.37	0.47	0.41	60
4	0.00	0.00	0.00	5
5	0.39	0.46	0.42	189
6	0.68	0.70	0.69	37
7	0.29	0.14	0.19	28
8	0.58	0.55	0.57	103
9	0.36	0.62	0.45	8
10	0.67	0.19	0.30	31
11	0.13	0.27	0.18	11
12	0.38	0.39	0.38	62
13	0.10	0.20	0.13	5
14	0.36	0.35	0.36	108
15	0.48	0.68	0.56	213
16	0.26	0.11	0.16	44
17	0.09	0.11	0.10	18
18	0.35	0.71	0.47	66
19	0.88	0.78	0.82	9
20	0.45	0.50	0.48	10
21	0.68	0.87	0.76	200
22	0.61	0.57	0.59	68
23	0.45	0.46	0.45	54
24	0.41	0.62	0.49	37
25	0.40	0.67	0.50	55
26	0.58	0.66	0.62	500
27	0.44	0.41	0.43	82
28	0.75	0.63	0.69	19
29	0.56	0.50	0.53	88
30	0.00	0.00	0.00	9
31	0.31	0.49	0.38	132
32	0.00	0.00	0.00	13
33	0.44	0.63	0.52	51
34	0.91	0.87	0.89	12184
35	0.50	0.47	0.48	30
36	0.56	0.69	0.62	26
37	0.33	0.50	0.40	2
38	0.36	0.08	0.13	52
39	0.41	0.52	0.46	264
40	0.60	0.74	0.66	346
41	0.26	0.21	0.23	148
accuracy			0.80	15509
macro avg	0.41	0.44	0.41	15509
weighted avg	0.81	0.80	0.81	15509

Table 10: BERT Classification Results on all data

BERT Overall Test Set Metrics on only relation data

- **Precision**: 0.6940
- **Recall**: 0.6956
- **F1-score**:0.6856

Class	Precision	Recall	F1-Score	Support
0	0.89	0.53	0.67	47
1	0.60	0.52	0.56	81
2	0.50	0.36	0.42	14
3	0.50	0.58	0.54	60
4	0.25	0.20	0.22	5
5	0.64	0.67	0.65	189
6	0.82	0.84	0.83	37
7	0.36	0.14	0.21	28
8	0.68	0.90	0.78	103
9	0.36	0.50	0.42	8
10	0.33	0.03	0.06	31
11	0.57	0.36	0.44	11
12	0.47	0.60	0.52	62
13	0.50	0.20	0.29	5
14	0.67	0.70	0.69	108
15	0.80	0.82	0.81	213
16	0.31	0.25	0.28	44
17	0.16	0.17	0.16	18
18	0.78	0.89	0.83	66
19	0.88	0.78	0.82	9
20	0.78	0.70	0.74	10
21	0.87	0.86	0.87	200
22	0.70	0.56	0.62	68
23	0.64	0.69	0.66	54
24	0.59	0.59	0.59	37
25	0.67	0.65	0.66	55
26	0.80	0.77	0.78	500
27	0.62	0.78	0.69	82
28	0.89	0.84	0.86	19
29	0.75	0.58	0.65	88
30	0.00	0.00	0.00	9
31	0.59	0.52	0.56	132
32	0.50	0.08	0.13	13
33	0.65	0.78	0.71	51
34	0.65	0.67	0.66	30
35	0.63	0.92	0.75	26
36	0.50	0.50	0.50	2
37	0.92	0.21	0.34	52
38	0.61	0.70	0.65	264
39	0.80	0.89	0.85	346
40	0.55	0.57	0.56	148
accuracy			0.70	3325
macro avg	0.60	0.56	0.56	3325
weighted avg	0.69	0.70	0.69	3325

Table 11: BERT Classification Report on Only Relations data

A.4 TACRED Dataset

Relation	Count
no_relation	84,491
per:title	3,862
org:top_members/employees	2,770
per:employee_of	2,163
org:alternate_names	1,359
per:age	833
per:countries_of_residence	819
org:country_of_headquarters	753
per:cities_of_residence	742
per:origin	667
org:city_of_headquarters	573
per:stateorprovinces_of_residence	484
per:spouse	483
org:subsidiaries	453
org:parents	444
per:date_of_death	394
org:stateorprovince_of_headquarters	350
per:children	347
per:cause_of_death	337
per:other_family	319
per:parents	296
org:members	286
per:charges	280
org:founded_by	268
per:siblings	250
per:schools_attended	229
per:city_of_death	227
org:website	223
org:member_of	171
org:founded	166
per:religion	153
per:alternate_names	153
org:shareholders	144
org:political/religious_affiliation	125
org:number_of_employees/members	121
per:stateorprovince_of_death	104
per:date_of_birth	103
per:city_of_birth	103
per:stateorprovince_of_birth	72
per:country_of_death	61
per:country_of_birth	53
org:dissolved	33

Table 12: All relation types in TACRED