Comparative Analysis Using SVM and BERT Models for Relation Extraction

Ismael Abidali Shaikh, Kavish Shah, Karthik Jayaprakash Menon, Gaurav Kewalramani
The University of Manchester

Abstract

Relation Extraction (RE) is a critical task in Natural Language Processing (NLP) that involves identifying the relationships between named entities in text. In this work, we present a Support Vector Machine (SVM)-based approach and a Bidirectional Encoder Representations from Transformers (BERT) based approach. The comprehensive evaluation highlighted the BERT's enhanced ability to discern complex relationships, surpassing the individual SVM model in precision, recall, and overall F1 score, thus affirming its superior performance in relation extraction.

1 Introduction

Relation Extraction (RE) is a fundamental task in NLP that aims to identify and classify the relationships between entities mentioned in text. This task is particularly important in domains such as biomedical literature, social media analysis, and knowledge graph construction. The Re-TACRED dataset, a revised version of the TACRED dataset, provides a rich source of annotated examples for relation extraction, making it an ideal choice for developing and evaluating RE methods.

In this paper, we explore the use of Support Vector Machines (SVM) and BERT for relation extraction on the Re-TACRED dataset. Our experiments demonstrate that while BERT-based models achieve performance on par with state-of-the-art models reported in the literature, SVM models, despite their simplicity, performed decent. This suggests that traditional machine learning methods, when combined with appropriate feature engineering, are overshadowed by deep learning methods for relation extraction tasks, especially in scenarios where computational resources are limited. We think that SVM ensemble stack would have been even better approach for a traditional ML task but due to computational limitation we were not able to compare it with SVM. To evaluate the performance of the proposed models, precision, recall, and F1 score metrics were employed and their results are discussed in the paper.

2 Dataset

For our task, we decided to use the updated version of the TACRED (Zhang et al., 2017) dataset i.e., Re-TACRED (Stoica et al., 2021). The primary reason for this decision was the updated annotations and resolved relation ambiguity in Re-TACRED. Some of the differences between the two datasets are highlighted below.

2.1 Original TACRED

The original TAC Relation Extraction Dataset (TACRED) is composed of over 106K examples and built upon newswire and web text, which is obtained from the Text Analysis Conference (TAC) Knowledge Base Population (KBP) challenges. Every example within the dataset consists of labeled subject and object entities within sentences. It consists of 41 relation types such as org:members (organization and its members), etc. Unfortunately, over time, it was discovered that it has several limitations, such as labeling noise, etc.

2.2 RE-TACRED

In 2021, (Stoica et al., 2021) built upon the TACRED dataset's limitations, such as a high error rate and relation ambiguity and conceived a significantly improved version namely 'Re-TACRED'. This was achieved by pruning poorly annotated sentences and resolving relation definition ambiguity. To achieve this feat, new crowd-sourced labels were obtained, which corrected 23.9% of the original labels from TACRED. Now, Re-TACRED consists of over 91K sentences and 40 relation types, which range from PERSON:CITY_OF_BIRTH to special NO_RELATION predicates, which indicate the absence of any relationship.

3 Literature Review

In this section, we perform a thorough critical analysis of past works based on the Re-TACRED dataset. More specifically, we focus on works using variations traditional machine learning and deep learning methods respectively i.e., SVM and BERT. We also define what SVM and BERT models are.

3.1 SVM Model

Support Vector Machines (SVM) classify data by finding the widest possible margin that separates classes (Cortes and Vapnik, 1995). rely on kernel transformations, which project data into higher-dimensional spaces to handle more complex decision boundaries. By focusing on margin maximization, SVMs often generalize well even with smaller datasets. However, combining SVMs into an ensemble can further enhance performance: multiple SVM models, each trained on varied subsets of the data or using slightly different parameter settings, can be aggregated—often by averaging or voting—to reduce variance and guard against overfitting. This ensemble approach helps capture diverse patterns that might be overlooked by a single SVM, leading to improved reliability and accuracy.

3.2 BERT Model

BERT (short for Bidirectional Encoder Representations from Transformers) is a neural network model designed to interpret text by looking at words in both directions—left and right—at the same time (Devlin et al., 2019a). It relies on a Transformer architecture that uses self-attention, allowing each word to gauge its relevance to every other word in a sentence. During pre-training, BERT employs two main strategies: masked language modeling (where it learns to predict intentionally hidden words) and next sentence prediction (where it judges whether one sentence follows another). This dual approach enables BERT to capture nuanced language patterns. After pre-training, the model

3.3 Related Works

The critical analysis of related work primarily pertains to transformer-based models i.e., BERT, since SVM usage on Re-TACRED is not as common. People have shifted towards deep-learning methods, therefore traditional methods gain almost no traction in certain areas.

To begin with, (Stoica et al., 2021) developed Re-TACRED by re-annotating it and evaluated the following models on the dataset; PA-LSTM, C-GCN models, and a transformer based model 'SpanBERT'. SpanBERT outperformed the other models tested on Re-TACRED, and emerged with an F1 score of 85.3%. This was a 5% increase over C-GCN and 5.9% increase over PA-LSTM. Moreover, it was a major jump from the earlier 69.7% F1 score achieved by SpanBERT on the TACRED dataset. A critical aspect to highlight is how the transformer based model performed better and had advantages over the other non-transformer models.

Another study completed by (Zhou and Chen, 2022) highlights the achievement of an F1 score of 91.1% through the use of typed entity markers and complete fine-tuning. They utilized a RoBERTa-large variant, combining it with a linear layer classification head. It was a straightforward approach with no advanced attention parameters or mixups, but highlighted the importance of entity markers, therefore achieving a very high score.

Furthermore, in a study by (Park and Kim, 2021), they built upon a BERT model, which was later fine-tuned. Following this, they incorporated a curriculum learning strategy, where training is done from easiest to most challenging i.e., the model can gradually adapt to complex instances. This proved to be beneficial since they achieved a micro-F1 score of approx. 91.4%, therefore setting a strong baseline further work.

3.4 Lessons Learned

There were quite a few lessons learned from the critical analysis of past work. Firstly, typed entity markers were of great benefit. They have proven to consistently help as shown in (Zhou and Chen, 2022). Followingly, fully fine-tuning proved to be a must thanks to the low noise dataset and high quality annotations. Moreover, techniques we did not apply such as external graphs, curriculum scheduling, etc., have proven to be able to push the usual F1 score further than the 90% mark. Based on this, our approach adopts some of the design elements and adds two unique/novel elements such as; a multihead, entity-aware attention from the [CLS] token, and Mixup specifically at the logits level, which is less typical for sentence-level RE.

4 Methodology

4.1 SVM

SVM is a type of machine learning algorithm that finds the optimal decision boundary (hyperplane)

between two classes of data while maximizing the margin between the classes. We used a linear SVM model for relation extraction. The SVM is trained on the TF-IDF vectors of the training set and evaluated on the development and test sets. The model is implemented using the scikit-learn library, with a linear kernel and a regularization parameter C=1.0. Firstly, we pre-processed the text data by converting all text to lowercase, removing punctuation, and eliminating extra spaces. The pre-processed text is then tokenized and converted into numerical features using TF-IDF vectorization. The TF-IDF vectorizer is configured to use unigrams and bigrams, with a maximum of 5000 features. We tried using more traditional ML approaches like random forest but it gave a lower accuracy as compared to SVM, we then also tried SVM ensemble stack which basically combines multiple classifiers for better performance. We used 3 different Support Vector Machine (SVM) models, each with a different kernel (linear, radial basis function (RBF), and polynomial) which basically served as base learners. These models individually analyse textual data using TF-IDF vectorized features. Rather than making final predictions directly, their probability outputs were stacked together as input for a meta-model, which in our case was a Logistic Regression classifier. But due to dataset size and time to compute and hardware constraints we were not able to get results from it, the model was running for more than 6 hrs and then google colab was giving prompt about runtime disconnecting. We believe that this model would have given us the best result for the traditional ML approach section.

4.2 Deep Learning with Transformers (BERT)

In this work, we incorporated and built upon a transformer-based encoder i.e., BERT, along with various enhancements and adjustments for performing relation extraction on the the Re-TACRED dataset. Some of the unique components are as follows;

- **Dynamic Entity Markers**: In our model, we encode the input sentences with markers around the subject and object entities. A bit of inspiration was obtained (Zhou and Chen, 2022), where their model performance increased due to the incorporation of entity markers. Our work adds these entity markers dynamically based on the entity span indices of the sentence. An example sentence would be "[John Smith] is married to [Hailey Smith]". Through the entity markers, this will be transformed into "[SUBJ_START] John Smith[SUBJ_END] is married to [OBJ_START] Hailey Smith [OBJ_END]". The typed markers allow the token itself to encode the type of entity, which allows the model to infer the semantic type constraints. Furthermore, we ensure that entity markers are not truncated by the 512-token limit of BERT during preprocessing.

- **Full Fine-Tuning**: We fine-tune our model fully i.e., all layers are unfrozen. We utilize the encoder as BERT-base uncased model. This is greatly beneficial especially when there are low-level features, thus helping in distinguishing relations. We also observed that the model did not overfit during the fine-tuning process. This is in line with observations from (Devlin et al., 2019b), where fine-tuning is said to yield better performance when sufficient data is provided.

- **Entity-Aware Attention Mechanism**: A novel component we introduced and have not observed in any paper is our entity-aware attention mechanism. Here, we have a multi-head attention layer on top of BERT. This allows the model to focus on the subjects and objects explicitly when it is tasked with forming a relational representation. This is done using the [CLS] representation as a query vector within a multi-head attention instance.

- **Mixup**: We also perform mixup on the logits, taking inspiration from image classification. This mixed relation training assists in reducing the phenomenon of overfitting and improves generalization.

- **Advanced Classification Head**: Our model also incorporates a denser and richer classification head, which consists of multiple linear-sub layers, LayerNorm, ReLU, and dropout mechanisms. This assists in addressing the class imbalance, since Re-TACRED has around 66% of all relations classed as 'no_relation'.

4.2.1 Architecture and Training

Below, we highlight the architecture and training details pertaining to our BERT model. We describe how it works and how it operates.

- **Architecture**: Firstly, the raw text is tokenized and based on our entity marking mechanism, markers are inserted around [subj] and [obj]'s. This is then sent to the encoder i.e., BERT. The classification head consists of multiple layers i.e., [Linear -> LayerNorm -> ReLU -> Dropout] x 2; Final Linear -> Output (num_labels).

- **Training**: We training the model upto 12 epochs on Re-TACRED's training set, with a batch size of 16. The base learning rate was 2e-5, and gradiant clipping was applied.

  When performing mixup, we randomly pick pairs of training examples with probability 0.5 at each step. We set $\alpha = 0.2$.

5 Evaluation

We evaluated the models using precision, recall, and F1-score metrics. These metrics provide a comprehensive assessment of the model's performance, particularly in the context of class imbalance.

SVM Model The SVM model achieved an F1-score of 0.57 on the training set and achieved an F1-score of 0.58 on the test set, suggesting that the use of TF-IDF features may not be sufficient to capture the semantic and contextual information needed for relation extraction. TF-IDF is a bag-of-words approach and does not consider word order or context, which are crucial for understanding relationships.

BERT Model Here, we follow the standard Re-TACRED splits: approx. 58k training instances, approx. 19.5k development instances, and approx. 13.4k test instances. Our main evaluation metric is micro-averaged F1, as it remains the most common measure in prior Re-TACRED work. We also report macro-F1 and a "filtered F1," which excludes the "no_relation" class from both predictions and references to measure how well the model performs on actual relation classes. After training for up to 12 epochs, our best checkpoint achieves:

- **Micro-F1:** 90.7% on the test set
- **Macro-F1:** 76.9%
- **Filtered-F1:** 79.9% (excluding the "no_relation" label)

We observe that misclassifications still occur for classes with only a few hundred training examples, such as "per:religion" or "per:stateorprovince_of_death. The advanced classification head kind of helps in mitigating these data-scarcity issues, but a direct label smoothing or specialized weighting might further help.

The BERT-based model significantly outperforms the SVM approach on the relation extraction task. While the SVM, relying on TF-IDF features, achieves modest F1-scores of approximately 0.57 on training and 0.58 on testing—limited by its bag-of-words representation that ignores word order and context—the BERT model leverages deep contextual embeddings and advanced mechanisms such as entity markers, entity-aware attention, and mixup training. These enhancements enable it to capture subtle semantic nuances and achieve a micro-F1 of around 90.7% on the test set. Overall, BERT's capacity for fine-tuning and understanding contextual dependencies translates into superior generalization and performance on the Re-TACRED dataset.

Limitations

There are a few limitations or shortcomings to the project. To begin with, the model requires a decent GPU to be trained on, which is a constraint if one cannot access such compute. But coming to BERT, we could have utilized a larger larger RoBERTa or BERT-large variant instead of bert-base-cased hich often yields +1–2% F1, and not combining multiple advanced methods such as label-graph networks or curriculum scheduling kind of lowered our F1 score a bit. Our training schedule with 12 epochs likely could be tuned further. Nevertheless, the model performed well.

One of the primary limitations of the SVM model is its inability to capture complex contextual information, which is crucial for accurate relation extraction. Additionally, the model's performance is heavily influenced by the quality of the TF-IDF features, which may not fully capture the semantic nuances of the text.

Ethics Statement

The work conducted in this paper complies with all ethical and government regulations.

References

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. 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 (NAACL-HLT)*, pages 4171–4186.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. Bert: Pre-training of deep bidirectional transformers for language understanding.

Seongsik Park and Harksoo Kim. 2021. Improving sentence-level relation extraction through curriculum learning.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-TACRED: Addressing shortcomings of the TACRED dataset. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 13843–13850.

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 (EMNLP 2017)*, pages 35–45.

Wenxuan Zhou and Muhao Chen. 2022. An improved baseline for sentence-level relation extraction. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 161–168, Online only. Association for Computational Linguistics.

A Appendix

A.1 Use of Generative AI

Generative AI was primarily utilized in the codebase aspect. This is included in the README.md file. In the report, AI was utilized to write down the mathematical function in LateX, and get directions for summaries.