Investigating the Performance of PCNN and BERT Models in Relation Extraction Tasks

Wei-Tung Lu
11356590

Tian Xu
11391352

Yihan Zhu
11360125

1 Introduction

In this project, we focus on utilizing the SemEval-2010 Task 8 dataset. This dataset, crafted by (Hendrickx et al., 2019), is a supervised training dataset known for its high quality and accuracy. It comprises 6,647 instances distributed across 9 classes.

Our work extends the OpenNRE framework (Han et al., 2019) , a versatile and open-source neural relation extraction toolkit developed by the Natural Language Processing Group at Tsinghua University (THUNLP). OpenNRE is renowned for its extensive collection of pre-trained models and its flexibility in training on customized datasets (Han et al., 2019).

To enhance model performance and user experience, we have specifically incorporated Convolutional Neural Network (CNN) and Bidirectional Encoder Representations from Transformers (BERT) models. These enhancements include:

1. CNN Model (Zeng et al., gust): We optimized the traditional CNN for the relation extraction task to handle classification problems. This model efficiently captures complex relational patterns between entities by learning local features within the text.

2. BERT Model: By leveraging the powerful language understanding capabilities of BERT, we have tailored a model for relation extraction (RE) tasks. This model captures the deep semantic relationships in texts through pre-trained bidirectional Transformer encoders, thereby improving the accuracy and robustness of relation extraction (Shi and Lin, 2019).

Our technical advancements in the project include:

- Resolving compatibility issues with new versions of the package, ensuring our modifications are fully compatible with the original OpenNRE codebase.

- Simplifying the configuration process to a one-click setup eliminates the need for extensive learning or manual effort to install and configure the environment.

- Enhancing the user experience by making it easier for users to utilize the model for relation extraction, thereby democratizing access to cutting-edge Natural Language Processing (NLP) tools.

By integrating CNN and BERT models, our project not only leverages the strengths of the OpenNRE framework but introduces innovations that make neural RE more accessible and efficient for researchers.

2 Related Work

In the field of NLP, particularly in the sub-task of RE, the integration of Word Embeddings, Position Embeddings, BERT, and CNN represents a significant advancement in understanding and extracting semantic relationships between entities within a text. Word Embeddings transform words into dense vector representations, capturing semantic meanings and enabling the model to process natural language in a computationally efficient manner (Kumar, 2017; Almeida and Xexéo, 2019). These embeddings serve as the foundational input for both BERT and CNN, facilitating the recognition of complex patterns and relationships in text data.

Along with word embedding, position embedding is another layer to encode the positional context of words within a sentence (Kumar, 2017). This is particularly crucial for tasks like RE, where the distance and order of words can significantly impact the nature of the relationship between entities. When combined with Word Embeddings, Position Embeddings allow models to understand not just what words are present, but also their relevance and relationship based on their positions.

BERT utilizes Word and Position Embeddings within its transformer architecture to deeply understand textual contexts. This model processes text bi-directionally, unlike traditional linear models, enabling a nuanced comprehension of each word's surrounding context (Devlin et al., 2018). Furthermore, BERT's transformer architecture significantly improves its capacity to capture complex linguistic patterns and dependencies, as it enables the model to discern the subtle interrelations between entities across diverse contexts.

CNN contributes to this ecosystem by applying convolutional layers to the text, enabling the extraction of local features and patterns. This capability aids the model in better comprehending the semantic information in sentences, enhancing its ability to accurately perform tasks like relation extraction.

These technologies have propelled the field of NLP and RE forward, offering nuanced insights into the semantic relationships between entities. The collaborative utilization of Word and Position Embeddings, along with the sophisticated architectures of BERT and CNN, exemplifies the dynamic nature of NLP research and its continuous pursuit of more refined, accurate, and efficient methods for processing natural language.

3 Methodology

3.1 CNN Based Model

The input to the relation extraction task consists of sentences that have been tokenised, and the model needs to determine the probability of these two entities belonging to each relationship based on the sentence and the specified two entities. Based on the requirement, we first chose the piecewise convolutional neural network (PCNN) (Zeng et al., 2015) implementation that incorporates the selective attention mechanism (Lin et al., 2016). One of the distinguishing features of PCNN is that it uses a piecewise max pooling layer. In general, the output Z of a convolutional layer is related to the size m of the input sentence. In order to make full use of the feature vectors in each dimension of a sentence, the max operation is used to collapse Z into a new two-dimensional tensor and thus decorrelate it with m (Kumar, 2017). It provides a better capture of the hierarchical information between entities whether comparing traditional CNNs or models with single-max pooling (Zeng et al., 2015). Before PCNN was proposed, relation extraction methods were either Feature-based or Kernel-based, and the effectiveness of these methods relied heavily on designing suitable feature sets (Zeng et al., 2015). PCNN has done better in the area of automatically learning features (Zeng et al., gust), hence reducing the reliance on manual tuning of the feature set. Consequently, the adoption of PCNN is justified by its provision of a nuanced feature representation, which concurrently bolsters the model's performance and interpretability.

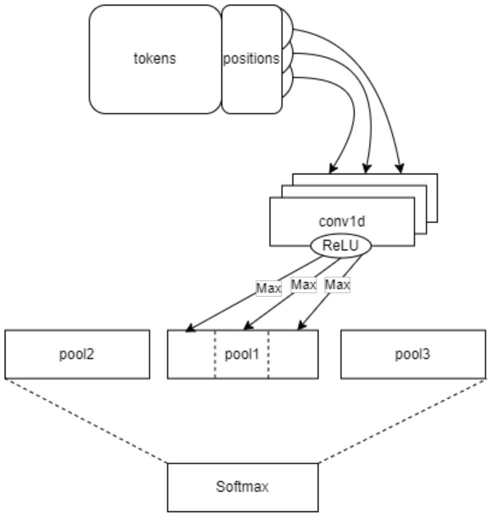

Figure 1: Illustration of the PCNN architecture.

As shown in Figure 1, the PCNN Encoder used in our project initializes with embeddings for tokens and positions, which are concatenated and then processed by a Conv1d layer. The network applies a ReLU activation function and piece-wise max pooling to capture the structural nuances around entities. A dropout layer is included for regularization, with the final output being a consolidated feature vector intended for sentence-level relation classification tasks.

We trained our own model using the chosen PCNN encoder. To train the model, we design the hyper-parameters as shown in Table 1 with reference to (Zeng et al., 2015). To accelerate the training process of the model on extensive datasets, we opted for an increased batch size. This decision was made in contrast to utilizing a smaller batch size of 50, which, while resulting in a 6.8% decrease in F1 score, facilitated a 44.7% improvement in training speed. To alleviate model over-fitting, the dropout probability was set to 0.5 to randomly activate the hidden layer. This probability will be set to 0 when performing the inference task to fully utilise the model's weights and features.

Window size	Feature maps	Word dimension	Position dimension	Batch size	Adadelta parameter	Dropout probability
3	230	$D_w = 50$	5	160	$\epsilon = 1e-5$, lr = 0.1	$p = 0.5$

Table 1: Training parameters for PCNN

3.2 Transformer based Model

In this project, Bidirectional Transformer Encoder Representation (BERT) is chosen as another approach for the relationship extraction task. Before going further, it is recommended to take a closer look at the BERT model itself.

The BERT is an NLP framework developed to initially train on a vast corpus of unlabeled text to learn deep, bidirectional language representations. Subsequently, it can be fine-tuned with labeled text across a variety of NLP tasks (Devlin et al., 2018). The architecture of the BERT model is a multi-layer, bidirectional Transformer encoder based on the implementation originally described in (Vaswani et al., 2017). The simplified BERT framework is shown in Figure 2.

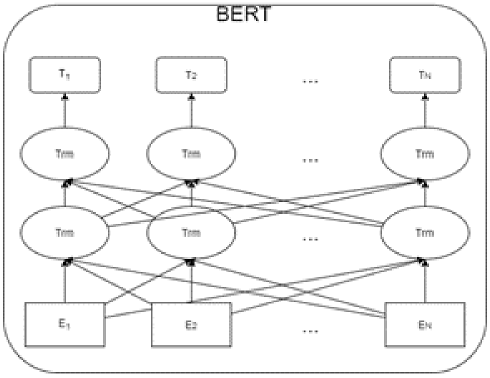

Figure 2: Simple structure diagram of BERT.

In the project, $E_1, E_2, \ldots, E_N$, represent the input embedding layer, which usually includes word embeddings and positional embeddings like WordPiece embeddings (Wu et al., 2016). These are representations of each word or token in the input sentence, and BERT treats them as part of the input sequence. Processing through multiple such Transformer encoder layers, each layer will process and refine the representation, allowing the model to capture deeper context, enhancing our model's accuracy and reliability in extracting relations. The transformers section has not been changed much and uses the existing transformers package directly. However, it is important to note that the version of the transformers package may cause problems with whether the model can be read or not, and in practice we noticed this effect of the version problem. The flow of how transformers work illustrates in Figure 3:

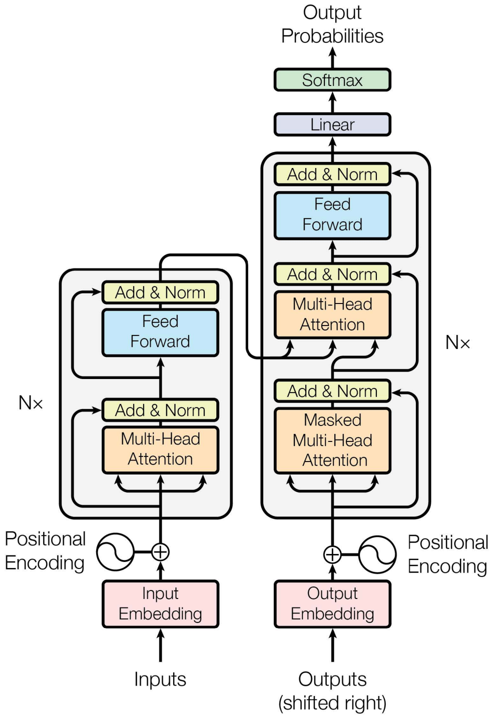

Figure 3: The architecture of transformer model(Vaswani et al., 2017).

Using the transformers package directly avoids a lot of mistakes, but some small details have been changed. As González-Carvajal and Garrido-Merchán suggest in their 2020 study (González-Carvajal and Garrido-Merchán, 2020), the `[MASK]` token was used in pre-training. Randomly "mask" (i.e., hide) a certain percentage of words (e.g., 15%) in the input text, and then let the model predict these masked words. This improves the model's ability to reason about entity relationships and makes it more accurate in dealing with implicit relationships.

In addition, `[entity]` tagging is used instead of

[CLS] tagging, allowing the model to focus more directly on the context around the entity, which can provide richer entity information for the relationship extraction task. [CLS] markup is typically used to aggregate global information, whereas not all global information is useful in relationship extraction. By not choosing this marker, the model can reduce the interference of irrelevant information and improve the accuracy of relationship extraction.

After pre-training is completed, the BERT model can be fine-tuned for specific downstream NLP tasks. The following pre-training model parameters and fine-tuning parameters are used for training in Table 2:

4 Evaluation Metrics

In evaluating our relation extraction approach, we adopted the metrics from (Han et al., 2019), crucial for assessing model efficacy in accurately identifying and classifying entity relationships. Our evaluation encompassed Accuracy, Micro Precision, Micro Recall, and Micro F1 Score.

Accuracy quantifies correct predictions' ratio to total instances, reflecting overall performance. However, its effectiveness may decrease with imbalanced datasets.

To mitigate accuracy's shortcomings in imbalanced datasets, Micro Precision and Micro Recall are employed. Both aggregate class-specific outcomes to produce comprehensive measures: Micro Precision by comparing all correctly predicted positive instances to all positive predictions, and Micro Recall by comparing true positives across classes to all actual positives.

The Micro F1 Score, harmonizing Micro Precision and Micro Recall, offers a balanced evaluation of precision and recall. It excels in assessing models where identifying all relevant instances and minimizing incorrect predictions are equally critical, penalizing false positives and negatives alike. Thus, the Micro F1 Score emerges as our preferred metric for its comprehensive and balanced approach.

5 Discussion

In evaluating RE models, the F1 score is vital as it balances precision and recall, crucial for class-imbalanced datasets.

The notebook's random selection of test set examples showed both PCNN and BERT accurately predicting outcomes. However, assessing overall test set performance offers a more comprehensive understanding.

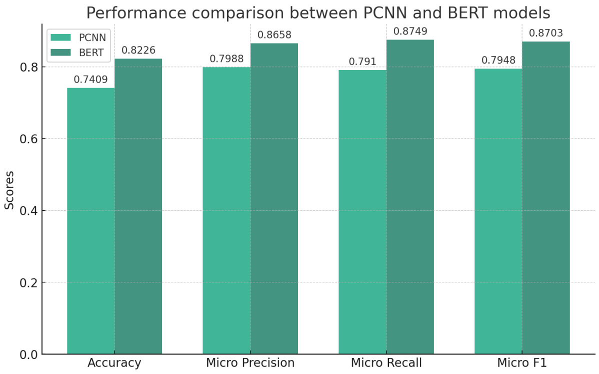

Figure 4: Performance comparison between PCNN and BERT models.

Figure 4 shows BERT outperforms PCNN in all metrics, thanks to its deep contextualized representations. BERT's accuracy is about 82.3% compared to PCNN's 74.1%. While accuracy is noted, the F1 score, balancing precision and recall, is more crucial, with BERT's F1 score exceeding PCNN's by roughly 8%.

Combining experimental insights and obtained data, the BERT and PCNN models exhibit distinct advantages and drawbacks. BERT's strengths include:

1. BERT captures long-distance dependencies and contextual information more effectively with its bi-directional Transformer structure, enhancing sentence semantics understanding. Its pre-training on extensive textual data imbues it with a broad linguistic knowledge base, aiding in complex linguistic phenomena comprehension and improving relation extraction accuracy, unlike PCNN's focus on local features.

2. The deep bi-directional structure and comprehensive pre-training grant BERT superior generalization across various domains and styles, whereas PCNN might need more domain-specific adjustments. BERT's use of sub-word units like WordPiece also enhances its handling of rare and polysemous words.

3. As an end-to-end model, BERT learns directly from raw text to relational labels without complex feature engineering, streamlining training and potentially boosting performance.

Pre-training Model	Layer	Hidden Layer Size	Self-attention Heads	Total Parameters	Batch size	Learning rate
BERT-base	12	768	12	approximately 110 million	64	2e-5

Table 2: Training parameters for Bert

As good as BERT is at RE tasks, it still has some flaws:

1. It demands significant computational resources for its deep architecture, unlike the simpler and less resource-intensive PCNN.
2. BERT's training, even for just 3 epochs, can match or exceed the duration needed for PCNN's 100 epochs, due to its larger parameter set, prolonging training and application times on vast datasets.
3. The model's complexity, stemming from numerous parameters and the intricate Transformer architecture, complicates debugging, optimization, and comprehension, in contrast to PCNN's simpler structure.

In conclusion, the choice between PCNN and BERT models for RE tasks depends on the specific requirements of the task at hand, the availability of computational resources, and the importance of contextual understanding. The F1 score serves as a comprehensive metric to evaluate these models, capturing both their precision and recall capabilities, which are essential for the balanced assessment of RE performance.

References

Almeida, F. and Xexéo, G. (2019). Word embeddings: A survey. *arXiv preprint arXiv:1901.09069*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

González-Carvajal, S. and Garrido-Merchán, E. C. (2020). Comparing bert against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012*.

Han, X., Gao, T., Yao, Y., Ye, D., Liu, Z., and Sun, M. (2019). Opennre: An open and extensible toolkit for neural relation extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 169–174.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Séaghdha, D. Ó., Padó, S., et al. (2019). Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. *arXiv preprint arXiv:1911.10422*.

Kumar, S. (2017). A survey of deep learning methods for relation extraction. *arXiv preprint arXiv:1705.03645*.

Lin, Y., Shen, S., Liu, Z., Luan, H., and Sun, M. (2016). Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133.

Shi, P. and Lin, J. (2019). Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, , and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zeng, D., Liu, K., Chen, Y., and Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762.

Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J. (2014, August). Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344.