BERT + SVM and BiLSTM with Relative Positional Encoding as approaches to Relation Extraction

Joshwin Sundarraj **Habiba Ahmad** **Dion Recai** **Manan Mal**

Abstract

In this paper, we delve into two RE methodologies: a BiLSTM-based model using Relative Positional Encoding and an SVM-based (BERT + SVM) model. The BiLSTM model captures the sequential dependencies in text. The BERT + SVM model, however, leverages pretrained contextual embeddings to boost relation classification. We train and evaluate both models on the SemEval dataset, applying metrics like accuracy, precision, recall, and F1 score to measure their effectiveness.

1 Introduction RE and Dataset

Extracting structured information from unstructured text has become a key challenge in Natural Language Processing (NLP). One of the core tasks in this area is Relation Extraction (RE), which focuses on identifying relationships between entity pairs within text. This task is vital for various real-world applications: building knowledge bases, developing automated Q&A systems, and mining information from biomedical texts. An RE needs to pinpoint the target entities within a given sentence. Following that, relation classification is applied to determine the relationships between contextual and surrounding words. This task can be quite complex, as relationships are often implicit, heavily reliant on context, and linguistically nuanced. Traditional ML approaches to RE utilise manually crafted features and syntactic rules. However, advancements in deep learning have allowed models to learn contextual representations directly from data, leading to improved generalization and performance.

For this project, we've selected the SemEval 2010 Task 8 as our benchmark dataset, specifically tailored for sentence-level relation extraction. SemEval stands out for its high-quality, human-annotated relations, which provide a reliable evaluation framework. This dataset has 19 directional relations including an "Other" category, offering a diverse array of relationships that challenge a model's ability to generalize. Due to its widespread use in NLP research, SemEval facilitates meaningful comparisons with existing state-of-the-art methods, making it an excellent choice for evaluating different approaches to RE (Hendrickx et al., 2010). A lot of work has been done surrounding relation extraction, with machine learning (ML) and deep learning (DL) approaches that leverage pretrained embeddings and contextual representations. Deep learning approaches, such as Transformers (Yang et al., 2021) and LSTMs (Miwa and Bansal, 2016), have demonstrated strong performance in extracting relations from text. On the other hand, traditional ML models including Support Vector Machines (SVMs) (Hong, 2005) and Naïve Bayes classifiers (Sureshkumar and Zayaraz, 2015), have also been widely applied, often relying on manually engineered features such as dependency parsing and word co-occurrence patterns.

2 Methodologies

2.1 BiLSTM approach

Long Short-Term Memory (LSTM) networks, are a type of Recurrent Neural Network (RNN) that has proven effective at capturing complex long-range dependencies (Zhang and Wang, 2015). Unlike standard RNNs, LSTMs mitigate against vanishing gradients via gated mechanisms that aid in regulating the flow of information over time and thereby preventing the loss of important contextual information (Zhang and Wang, 2015). BiLSTMs extend this notion by respectively using a pair of LSTMS to capture context in both the forward and backward direction when processing sequences.

2.1.1 Sample Pre-processing

Sample sentences in the dataset are tagged with positional indicators (e.g <e1>,</e1>,<e2>,</e2>). Experimental findings by (Baldini Soares et al., 2019) (Tao et al., 2019) indicate that maintaining these "syntactic" indicators improves the performance hence they were not filtered out. Minor preprocessing is done to transform the indicators by separating them by space to ensure proper tokenisation. Punctuation besides: <,>, \ are filtered out to reduce noise/OOV tokens (out-of-vocabulary) since they can introduce unnecessary complexity during tokenisation. Finally, tokenised sequences are padded to ensure inputs to the BiLSTM are uniform. Additionally, relation labels associated with each sentence sample are one-hot encoded to a vector of size 19 (no. relation classes).

2.1.2 Relative Positional Encoding

Positional encoding techniques provide models with an extra layer of context about relative and absolute positions of words in sequences. Whilst LSTMs and BiLSTMs implicitly capture positional relationships, explicit positional encoding enhance the model's ability to distinguish between different relationships in relation classification. This has been demonstrated in research by (Bilan and Roth, 2018) indicating that relative positional encoding combined self-attention mechanism improves a model's performance in relation extraction. Consequently, applying this technique would improve the model's ability to recognize sequence-order dependencies in tasks like relation extraction which is also demonstrated in research by (Shaw et al., 2018). In our method, we take inspiration from (Bilan and Roth, 2018) make use of a much simpler bespoke encoding technique in which two position encodings are calculated based on the relative distance of words in a sample sentence both tagged entities respectively. For example, the sentence "The <e1>bottle</e1> was filled with <e2>coloured water</e2> and placed on the table." would produce produce positional encodings w.r.t <e1>bottle</e1> ($[-2, -1, 0, +1, ..., +13]$) and <e2>water</e2> ($[-8, ..., -1, 0, 0, +1, ..., +6]$), such that tagged entities are masked with 0. Like the tokenised sequences produced, these positional encodings are padded to the same lengths.

2.1.3 BiLSTM Architecture

BiLSTM architecture prepared for this approach consisted of an embedding layer where pre-trained GloVe embeddings are used to initialize a word embedding matrix (Pennington et al., 2014). Input tokenised sequences are mapped to a dense vector representation, the size of which is a hyperparameter. Position embeddings are also generated from the position encoding inputs. The three embeddings are then concatenated and passed through a 1D spatial dropout layer. Dropout is applied since research indicates that randomly deactivating neurons reduces over-fitting and thereby increase generalisability (Gal and Ghahramani, 2016) (Srivastava et al., 2014) (Zaremba et al., 2014). Following on, there is a single BiLSTM layer with: no. of hidden units, dropout value and l2-regularisation value, all of which are hyper-parameters to be tuned.

A single layer was utilised since research by (Peters et al., 2018) demonstrated that lower-layers in BiLSTMs capture significant contextual and syntactic information, implying strong performance be achieved with fewer layers, thereby reducing computational costs. Finally, relation classification is performed through a dense layer with softmax activation which produced probability distributions over the 19 relation types.

2.1.4 Bayesian Hyperparameter Optimisation

Given deep learning techniques are being used to train the model, using a Grid Search based algorithm would simply take too long to find optimal hyperparameters. Therefore we opted to utilise Bayesian Optimisation to effectively, compute optimal hyperparameters for the model. Applying this strategy provided the following optimal values for our hyper-parameter. Best Hyperparameters: {'embedding dim': 300, 'lstm units': 96, 'dropout rate': 0.8, 'learning rate': 0.001, 'bilstm dropout': 0.4, 'kernel regularizer': 0.0001}

2.2 BERT + SVM Approach

2.2.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained transformer model that can be fine-tuned to perform various language-related tasks such as masked language modeling, next sentence prediction, and question answering (Devlin et al., 2019). To accomplish this and facilitate fine-tuning, BERT converts each token in an input sequence into an embedding vector that captures the context of the token and the sentence as a whole (Devlin et al., 2019). The base BERT model's output embedding vectors have 768 features (Devlin et al., 2019). We decided to use the base model because the large model requires a lot more computing power and performs only marginally better (Shi and Lin, 2019). BERT's embedding vectors on their own (without further fine-tuning) have been shown to be useful for relation extraction due to the context that they capture in their features (Shi and Lin, 2019) (Wang et al., 2022). Another model can then use the BERT embeddings to perform relation extraction; we propose using an SVM.

2.2.2 SVM

We chose to implement a Support Vector Machine model, which is called a Support Vector Classifier (SVC) when used for classification, due to its resilience to common problems of other models, its computational efficiency, its advanced performance in classification tasks, and its robustness in cases of overfitting (Hong, 2005).

Support Vector Machines (SVM) work by finding a hyperplane that best separates different classes within a given feature space (Noble, 2006).

SVMs have previously been employed to extract relations in clinical records between entities i.e conditions, drug and result) (Minard et al., 2011) as well as classifying relations between entities in languages such as Spanish (Torres et al., 2018). The latter was enhanced, having added contextual and syntactic features which improve classification tasks. Additionally, SVM classification models have been created within a medical context to extract embeddings from clinical features (Khan et al., 2024). The model had an accuracy of 95%, highlighting its success in classification, suggesting that this approach can be extended to RE (since relation classification is a subset of this).

2.2.3 Combining BERT + SVM

To train an SVM model, each data instance must be represented as a vector of features. Our pipeline uses BERT to produce sentence embedding vectors, and then uses an SVM to classify the embeddings into one of the relation types.

The input sentences in the SemEval dataset contain tags to indicate the location of the entities. Previous research (Baldini Soares et al., 2019) as well as our own experimentation has shown that including tags in the sentence produces better relation learning results, so we decided to keep the entity tags. BERT requires the input sentence to be tokenised using an encoding system known as WordPiece (Devlin et al., 2019). We pass the encoded tokens into the model to produce the embedding vectors. We use the transformers PyPI package for the tokeniser and the BERT model itself. Since the SVM needs a single vector to represent the input sentence, we used the mean of the embedding vectors of all tokens in the sentence as the sentence embedding vector. A common approach is to use the embedding vector for the sentence's [CLS] token, which is a special token that WordPiece includes at the start of the token sequence; however, prior research (Reimers and Gurevych, 2019) and our own experimentation show that using the mean can lead to better performance. We experimented with using PCA to reduce the number of features but the results were significantly worse. The embedding vectors are used to train the SVM model.

2.2.4 SVM Hyperparameter Tuning

GridSearchCV takes the SVC object along with a dictionary of hyperparameters to be tested during grid search which performs an exhaustive search over the parameter space — considering all combinations — to search for the best parameters. We set the kernel function parameter as linear which is recommended for text classification (Hsu et al., 2009). We also tuned class_weight and parameter C which can be used to obtain a higher level of generalization (Guo and Wang, 2015). Cross-validation is said to be the best approach for SVM parameter selection (Gholami and Fakhari, 2017). In our case, we carried out cross-validation with 5 folds.

3 Results of Evaluation and Discussion

	F1 Score	Final Accuracy
BiLSTM RPE	70.42	75.27
BERT + SVM	59.0	64.26

Table 1: Final test accuracy and macro-average F1 score for both approaches

The test accuracy and F1 score were metrics observed when evaluating the 2 aforementioned approaches. Accuracy was used primarily because it is easy to understand, while F1 scores were also observed to further evaluate how well our approaches handle false positives and false negatives. In addition, precision, recall, and F1 scores were observed for each type of relation as seen in Table 2 within the appendix. The combination of all these metrics were observed to increase our confidence in the experimental results. Given how the data exist as distinct train and test sets, the final evaluation on the BiLSTM approach was done by training the model (over 30 epochs of batch sizes of 10) on the entire training set using optimal hyperparameters observed from Bayesian optimization. Since this method has shown to produce parameters that help models generalize well, there was no need for Cross-Validation in this instance. The BERT + SVM approach was evaluated in a similar fashion.

Results indicate that the single layer BiLSTM + RPE outperforms BERT + SVM on all metrics observed regardless of augmenting SVM with embeddings produced by BERT (please refer to 2 in the appendix). This was expected since most of the relation classification in the latter was done by an SVM, unlike the former approach. In addition, whilst BiLSTM + RPE was observed to have strong performance, it did not perform as well as the SOTA BiLSTM model demonstrated in (Zhou et al., 2016), which had an accuracy of 85% over the SemEval dataset. This can be attributed to the fact the paper makes use of a unique attention mechanism.

An interesting point to note is that both the BiLSTM + RPE and BERT + SVM got 0% on all metrics for the Entity-Destination(e1,e1) relation. This is likely because there was only one training point for that relation, so the models were unable to learn to predict that relation. Some methods that could be used to deal with this issue are oversampling, such as SMOTE (Chawla et al., 2002) (Satriaji and Kusumaningrum, 2018), converting sentences from the reverse relation to this one, or simply getting rid of this relation from the data.

SVM classifiers tend to provide better precision than recall scores for relation extraction (Detroja et al., 2023), but this did not occur with our BERT + SVM model, as we obtained a precision of 59% and a recall of 60% (2).

SVMs using specialised kernels and entity and POS tagging features tend to have good precision scores compared to our model, as they are usually in the 63% - 84% range, while our model produces a final precision of 59%. Our model however has a recall of 60% and an F1 of 59% (2), which is better than approximately half of these models, whose recall scores are between 26% - 93% and F1 scores between 38% to 86% (Detroja et al., 2023). This shows that the context provided by BERT embeddings can be comparable to specialised kernels and tagging. One SVM model by (Rink and Harabagiu, 2010) used the same SemEval dataset as us and achieved 82.19% — which was significantly higher than ours — by iteratively adding more tagging features such as WordNet, NomLex-Plus7, VerbNet, and the Google N-Gram.

Using BERT embeddings for other models, especially sequential models like BiLSTMs, or other fine-tunings of BERT, produce better F1 scores than our model which simply uses an SVM. For example, the F1 scores of these models are: 67.8% (Shi and Lin, 2019), 95.02% (Wang et al., 2022), 89.2% (Baldini Soares et al., 2019). (Wu and He, 2019) achieved 89.25% F1 score using the same SemEval dataset as us. This shows that using BERT to capture context may not be sufficient for an SVM to match the performance of LSTMs and transformers.

4 Conclusion

In this study, we explored and compared two distinct approaches for Relation Extraction (RE) on the SemEval 2010 Task 8 dataset: a BiLSTM-based model with Relative Positional encoding and a BERT + SVM hybrid model. Our experimental results highlight several key findings. Firstly, the single-layer BiLSTM model outperformed the traditional SVM across all evaluation metrics, demonstrating the effectiveness of deep learning in capturing sequential dependencies and contextual relationships. This reinforces the advantage of end-to-end feature learning over manually engineered features used in traditional machine learning models. Secondly, the BERT + SVM model exhibited competitive performance, surpassing the BiLSTM model in terms of precision for few relation types. However, it showed room for improvement, particularly in recall and overall robustness. Overall, our findings indicate that deep learning models such as BiLSTM with RPE provide better performance for relation extraction, even after augmenting SVM with BERT embeddings. Though, the latter shows promise and has room for improvement when enhanced with richer feature representations (Rink and Harabagiu, 2010). Future work could explore directly fine-tuning BERT or integrating graph-based approaches to further enhance relation extraction capabilities.

References

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Ivan Bilan and Benjamin Roth. 2018. Position-aware self-attention with relative positional encodings for slot filling. *ArXiv*, abs/1807.03052.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Kartik Detroja, C.K. Bhensdadia, and Brijesh S. Bhatt. 2023. A survey on relation extraction. *Intelligent Systems with Applications*, 19:200244.

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.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Raoof Gholami and Nikoo Fakhari. 2017. Chapter 27 - support vector machine: Principles, parameters, and applications. In Pijush Samui, Sanjiban Sekhar, and Valentina E. Balas, editors, *Handbook of Neural Computation*, pages 515–535. Academic Press.

Husheng Guo and Wenjian Wang. 2015. An active learning-based svm multi-class classification model. *Pattern Recognition*, 48(5):1577–1597.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Gumwon Hong. 2005. Relation extraction using support vector machine. In *Natural Language Processing – IJCNLP 2005*, pages 366–377, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. 2009. A practical guide to support vector classification.

Asma Sadia Khan, Fariba Tasnia Khan, Tanjim Mahmud, Salman Karim Khan, Nahed Sharmen, Mohammad Shahadat Hossain, and Karl Andersson. 2024. Integrating bert embeddings with svm for prostate cancer prediction. In *2024 6th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, pages 01–06.

Anne-Lyse Minard, Anne-Laure Ligozat, and Brigitte Grau. 2011. Multi-class SVM for relation extraction from clinical reports. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 604–609, Hissar, Bulgaria. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

William S Noble. 2006. What is a support vector machine? *Nature Biotechnology*, 24.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Bryan Rink and Sanda Harabagiu. 2010. UTD: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 256–259, Uppsala, Sweden. Association for Computational Linguistics.

Widi Satriaji and Retno Kusumaningrum. 2018. Effect of synthetic minority oversampling technique (smote), feature representation, and classification algorithm on imbalanced sentiment analysis. In *2018 2nd International Conference on Informatics and Computational Sciences (ICICoS)*, pages 1–5.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* *Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Peng Shi and Jimmy J. Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

G. Sureshkumar and G. Zayaraz. 2015. Automatic relation extraction using naïve bayes classifier for concept relational ontology development. *International Journal of Computer Aided Engineering and Technology*, 7(4):421–435.

Qiongxing Tao, Xiangfeng Luo, Hao Wang, and Richard Xu. 2019. Enhancing Relation Extraction Using Syntactic Indicators and Sentential Contexts . In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1574–1580, Los Alamitos, CA, USA. IEEE Computer Society.

Jefferson Peña Torres, Raúl Gutierrez de Piñerez Reyes, and Víctor A. Bucheli. 2018. Support vector machines for semantic relation extraction in spanish language. In *Advances in Computing*, pages 326–337, Cham. Springer International Publishing.

Xiaoguo Wang, Yanning Sun, Chao Chen, and Jianwen Cui. 2022. A relation extraction model based on bert model in the financial regulation field. In *2022 2nd International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)*, pages 496–501.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 2361–2364, New York, NY, USA. Association for Computing Machinery.

Xi Yang, Zehao Yu, Yi Guo, Jiang Bian, and Yonghui Wu. 2021. Clinical relation extraction using transformer-based models.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *ArXiv*, abs/1409.2329.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *ArXiv*, abs/1508.01006.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany. Association for Computational Linguistics.

A Appendix

	BiLSTM + Relative Positional Encoding			BERT + SVM		
Relation	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
Cause-Effect(e1,e2)	0.89	0.89	0.89	0.85	0.63	0.72
Cause-Effect(e2,e1)	0.89	0.87	0.88	0.63	0.88	0.74
Component-Whole(e1,e2)	0.80	0.78	0.79	0.68	0.70	0.69
Component-Whole(e2,e1)	0.62	0.64	0.63	0.51	0.65	0.57
Content-Container(e1,e2)	0.72	0.95	0.82	0.79	0.83	0.81
Content-Container(e2,e1)	0.78	0.74	0.76	0.56	0.56	0.56
Entity-Destination(e1,e2)	0.85	0.90	0.87	0.73	0.85	0.78
Entity-Destination(e2,e1)	0.00	0.00	0.00	0.00	0.00	0.00
Entity-Origin(e1,e2)	0.79	0.87	0.83	0.59	0.78	0.67
Entity-Origin(e2,e1)	0.80	0.87	0.84	0.65	0.77	0.71
Instrument-Agency(e1,e2)	0.52	0.55	0.53	0.33	0.27	0.30
Instrument-Agency(e2,e1)	0.69	0.73	0.71	0.69	0.48	0.56
Member-Collection(e1,e2)	0.43	0.66	0.52	0.50	0.41	0.45
Member-Collection(e2,e1)	0.80	0.91	0.85	0.63	0.88	0.73
Message-Topic(e1,e2)	0.76	0.90	0.82	0.71	0.80	0.75
Message-Topic(e2,e1)	0.78	0.75	0.76	0.78	0.57	0.66
Product-Producer(e1,e2)	0.76	0.78	0.77	0.60	0.58	0.59
Product-Producer(e2,e1)	0.68	0.69	0.69	0.55	0.53	0.54
Other	0.54	0.33	0.41	0.49	0.22	0.30
Macro Avg	0.69	0.73	0.70	0.59	0.60	0.59
Weighted Avg	0.74	0.75	0.74	0.63	0.64	0.62

Table 2: Metrics per relation type in relation extraction over unseen data for both approaches