Understanding Context from Relation Extraction on SemEval 2010: Insights from Machine Learning and Deep Learning Approaches

COMP61332 Text Mining Coursework - The University of Manchester

Patricio Jaime Porras
patricio.jaimeporras
@postgrad.manchester.ac.uk

Radoslaw Izak
radoslaw.izak
@postgrad.manchester.ac.uk

Roshan Bhalaji
Ramesharavind Lathamahesh
roshan.bhalaji
@postgrad.manchester.ac.uk

***Abstract*—Relation extraction is a fundamental task in Natural Language Processing (NLP) that aims to determine the relationships between named entities in a text. In this study, we compare the effectiveness of deep learning architectures, specifically Long Short-Term Memory (LSTM) networks, and conventional machine learning algorithms, including Support Vector Machines (SVM) and Naive Bayes, for Relation Extraction. Additionally, we explore the use of pre-trained transformers, such as BERT, for generating embeddings to capture contextual information and enhance performance. Our experiments are conducted on the SemEval 2010 dataset (task 8), which contains annotated instances for Relation Extraction. We evaluate the performance of the developed approaches and compare them against each other as well as how they can be enhanced. The results provide insights into the strengths and limitations for each technique used and give significant implications for improving the understanding of semantic relations in text.**

***Index Terms*—NLP, Relation Extraction, Text Mining, Machine Learning**

I. Introduction

The main aim of this report is to improve relation extraction techniques by developing, evaluating, and comparing two main Relationship extraction methods applied to the SemEval 2010 Task 8 dataset. The three main objectives of this report are:

1) Build two main RE approaches and fit the models with the chosen dataset.
2) Evaluate and compare these methods while applying novel techniques to make them better at predicting the relationships between the chosen entities.
3) Give a detailed and in-depth analysis of the RE methodologies.

The main foundation of our study is the SemEval dataset which offers a very good collection of textual instances with 18 named entity relationships (19 for *"Other"*). By using this well-established dataset, a comparison between our results and the broader context of relation extraction research can be made.

This study explores the complexities of developing Relation Extraction techniques using two approaches on the SemEval dataset. The first relies on traditional machine learning methods, Support Vector Machines (SVM) and Naive Bayes, while the second utilizes deep learning, specifically Long Short-Term Memory (LSTM) networks. By comparing these approaches, we aim to gain insights into their strengths, weaknesses, and effectiveness. Our findings provide valuable insights and guide future research efforts in this area of NLP, aspiring to lay a foundation for more sophisticated and efficient methods/models.

II. Related Work

Our study focuses on developing and comparing two distinct RE approaches applied to the SemEval 2010 dataset. The insights gained from prior studies have greatly influenced our research. By examining the strengths and weaknesses of traditional and deep learning paradigms, this report contributes to the ongoing evolution of RE methodologies.

Relation Extraction (RE) has gained significant attention in natural language processing, leading to the exploration of various methodologies and techniques. This section provides an analysis of relevant studies, offering a contextual background for our research.

A. Traditional ML Approaches

Early RE methods used traditional machine learning techniques, such as Support Vector Machines (SVMs) and Naive Bayes. Hong (2005) made use of SVMs to identify informative features and developed an experimental process to enhance RE performance. These approaches laid the foundation for RE research.

B. Deep learning approaches

With the developing of deep learning, Relation Extraction underwent a paradigm shift. Researchers began employing neural networks to capture complex linguistic properties. Zhang et al. (2015) introduced the Bidirectional Long Short-Term Memory (Bi-LSTM) networks for relation classification, demonstrating their ability to model sequential dependencies. Recent work by Thillagaisundaram and Togia (2019) integrated pre-trained language models with minimal task-specific architectures, showcasing the evolving trends in deep learning for RE.

C. *Transformer-based Approaches (Attention is All You Need)*

The introduction of the Transformer architecture by Vaswani et al. (2017) revolutionized NLP tasks, including RE. Transformers utilize self-attention mechanisms to capture long-range dependencies and contextual information effectively. Models like BERT (Devlin et al., 2019) have achieved state-of-the-art performance in RE by leveraging pre-trained Transformer encoders. These models can be fine-tuned on RE datasets, enabling them to learn task-specific representations.

D. *Datasets*

Dataset selection is crucial in RE research. The SemEval 2010 Task 8 dataset has become a widely adopted benchmark for evaluating RE approaches. Additionally, the FewRel, another notable dataset, has facilitated advancements in few-shot RE. These datasets enable the comparison and evaluation of different RE methods.

III. Methodologies

As discussed previously, we started by looking into the traditional machine learning algorithms, followed by the use of deep learning algorithms. For both of this methods, we used a series of techniques to try to improve the model's performance on the same dataset (SemEval 2010 - task 8). Our approach is structured in a way that encompasses both traditional machine learning and deep learning paradigms to ensure a comprehensive exploration of RE methodologies.

A. *Traditional Approaches*

For the the traditional methods we employ well-established traditional machine learning algorithms like Support Vector Machines (SVMs) and Naive Bayes for relation extraction. SVMs are known for their ability to find optimal hyperplanes in high-dimensional feature spaces while Naive Bayes classifiers are probabilistic classifiers that can often outperform more sophisticated classification methods while still being very simple (Vijaykumar B et al. 2014). These algorithms have been widely used in RE research due to their simplicity and robustness. We leverage linguistic and syntactic features, such as dependency relations and named entity types to train these models as well as the use of BERT embeddings. By comparing the performance of SVMs and Naive Bayes, we aim to establish a baseline of traditional machine learning approaches in relation extraction.

1) Naive Bayes: For Naive Bayes, we tested many of its variants, such as Gaussian-NB, Bernoulli-NB, Multinomial-NB, and Complement-NB with vectorization of words using bag-of-words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and BERT embeddings. Overall, Complement Naive Bayes created a more performant model, as expected since this variant of NB is suited to work with imbalanced datasets and calculates the probabilities for every class. Results can be seen in TABLE I.

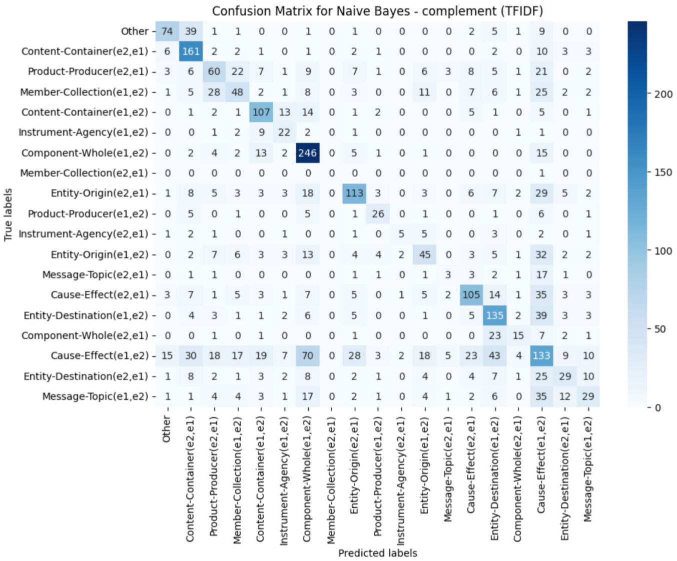

Fig. 1. Confusion Matrix for Complement NB with TF-IDF

Relation	Precision	Recall	F1-score	Support
Cause-Effect(e1,e2)	0.70	0.55	0.62	134
Cause-Effect(e2,e1)	0.57	0.83	0.68	194
Component-Whole(e1,e2)	0.43	0.37	0.40	162
Component-Whole(e2,e1)	0.41	0.32	0.36	150
Content-Container(e1,e2)	0.61	0.70	0.65	153
Content-Container(e2,e1)	0.37	0.56	0.45	39
Entity-Destination(e1,e2)	0.58	0.85	0.69	291
Entity-Destination(e2,e1)	0.00	0.00	0.00	1
Entity-Origin(e1,e2)	0.62	0.54	0.57	211
Entity-Origin(e2,e1)	0.59	0.55	0.57	47
Instrument-Agency(e1,e2)	0.50	0.23	0.31	22
Instrument-Agency(e2,e1)	0.43	0.34	0.38	134
Member-Collection(e1,e2)	0.21	0.09	0.13	32
Member-Collection(e2,e1)	0.61	0.52	0.56	201
Message-Topic(e1,e2)	0.51	0.64	0.57	210
Message-Topic(e2,e1)	0.48	0.29	0.37	51
Other	0.30	0.29	0.30	454
Product-Producer(e1,e2)	0.41	0.27	0.32	108
Product-Producer(e2,e1)	0.41	0.24	0.30	123
accuracy			0.50	2717
macro avg	0.46	0.43	0.43	2717
weighted avg	0.49	0.50	0.48	2717

TABLE I
PERFORMANCE METRICS FOR COMPLEMENTNB WITH TF-IDF

Regarding the vectorization, the performance of BoW and TF-IDF is quite comparable. However, BERT *embeddings* negatively impact the classifier's performance (TABLE II)., which is not surprising given that Naive Bayes classifiers, such as Complement-NB, work under the assumption of feature independence. This assumption make BERT embeddings ineffective for capturing contextual information. To improve the classifier's performance with BERT embeddings, fine-tuning both BERT and the model would be necessary. This could involve implementing a more appropriate kernel or reducing the dimensionality of the embeddings to minimize noise from the BERT representations.

2) SVM: For the SVMs, Support Vector Classification, the performance was not very dissimilar to Naive Bayes. The best result came by using 'small-bert' instead of 'tiny-bert' giving the highest accuracy of all the traditional approaches

Relation	Precision	Recall	F1-score	Support
Cause-Effect(e1,e2)	0.00	0.00	0.00	134
Cause-Effect(e2,e1)	0.35	0.68	0.46	194
Component-Whole(e1,e2)	0.38	0.33	0.35	162
Component-Whole(e2,e1)	0.52	0.17	0.25	150
Content-Container(e1,e2)	0.42	0.54	0.47	153
Content-Container(e2,e1)	0.00	0.00	0.00	39
Entity-Destination(e1,e2)	0.27	0.82	0.41	291
Entity-Destination(e2,e1)	0.00	0.00	0.00	1
Entity-Origin(e1,e2)	0.53	0.04	0.07	211
Entity-Origin(e2,e1)	0.56	0.19	0.29	47
Instrument-Agency(e1,e2)	0.00	0.00	0.00	22
Instrument-Agency(e2,e1)	0.44	0.03	0.06	134
Member-Collection(e1,e2)	0.00	0.00	0.00	32
Member-Collection(e2,e1)	0.23	0.79	0.36	201
Message-Topic(e1,e2)	0.42	0.56	0.48	210
Message-Topic(e2,e1)	0.00	0.00	0.00	51
Other	0.31	0.03	0.05	454
Product-Producer(e1,e2)	0.33	0.01	0.02	108
Product-Producer(e2,e1)	0.50	0.03	0.06	123
accuracy			0.31	2717
macro avg	0.28	0.22	0.17	2717
weighted avg	0.34	0.31	0.23	2717

TABLE II
PERFORMANCE METRICS FOR COMPLEMENTNB WITH SMALL BERT

Relation	Precision	Recall	F1-score	Support
Cause-Effect(e1,e2)	0.88	0.53	0.66	134
Cause-Effect(e2,e1)	0.65	0.73	0.69	194
Component-Whole(e1,e2)	0.60	0.54	0.57	162
Component-Whole(e2,e1)	0.52	0.32	0.40	150
Content-Container(e1,e2)	0.70	0.77	0.73	153
Content-Container(e2,e1)	0.92	0.31	0.46	39
Entity-Destination(e1,e2)	0.68	0.79	0.73	291
Entity-Destination(e2,e1)	0.00	0.00	0.00	1
Entity-Origin(e1,e2)	0.64	0.49	0.56	211
Entity-Origin(e2,e1)	0.69	0.47	0.56	47
Instrument-Agency(e1,e2)	0.00	0.00	0.00	22
Instrument-Agency(e2,e1)	0.47	0.22	0.30	134
Member-Collection(e1,e2)	0.00	0.00	0.00	32
Member-Collection(e2,e1)	0.63	0.66	0.65	201
Message-Topic(e1,e2)	0.61	0.70	0.65	210
Message-Topic(e2,e1)	0.92	0.22	0.35	51
Other	0.26	0.52	0.35	454
Product-Producer(e1,e2)	0.70	0.06	0.12	108
Product-Producer(e2,e1)	0.76	0.11	0.19	123
accuracy			0.52	2717
macro avg	0.56	0.39	0.42	2717
weighted avg	0.58	0.52	0.51	2717

TABLE III
PERFORMANCE METRICS FOR SVC WITH SMALL BERT

(TABLE III). SVMs outperforming NB is also expected due to SVMs having non-linear decision boundaries rather than linear and that they do not make the assumption of feature independence. Our only issue with SVMs is that they took to long to compute without the use of BERT. Delving into these issue, it is probably because both BoW and TF-IDF often result in high-dimensional matrices that include many zero values. This matrices can become really hard to use while computing a vast amount of words with the '*rbf*' or any non-linear kernel. We can not imply that SVMs with BERT will outperform other vectorization techniques without **attention**.

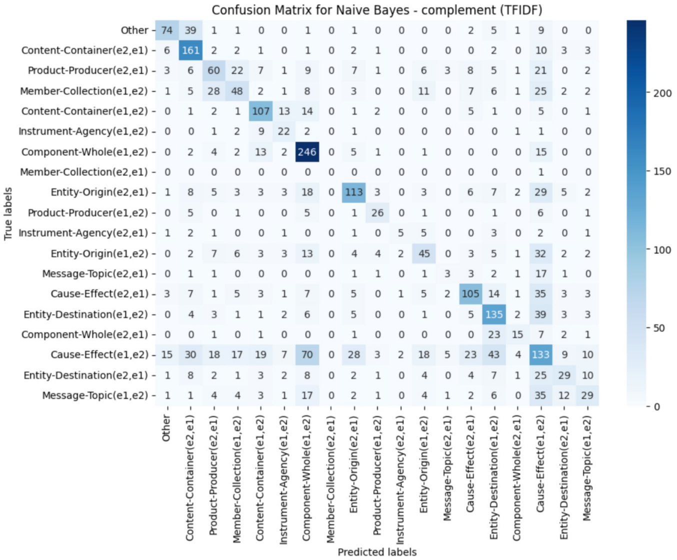

Fig. 2. Confusion Matrix for SVM with Small BERT

B. Deep Learning Approach

In the deep learning approach, we selected the Bidirectional Long Short-Term Memory (BiLSTM) networks, a type of recurrent neural network that is very useful at capturing sequential dependencies in textual data. We utilized BiLSTM to learn complex patterns and understand the relationships within the SemEval 2010 Task 8 dataset. Furthermore, with the use of BERT and GloVe, LSTMs provide rich word embeddings which gives the model a vast knowledge of unlabeled data. This improves the model performance on tasks such as relation extraction. Our results can be seen on TABLE IV.

Relation	Precision	Recall	F1-score	Support
Cause-Effect(e1,e2)	0.90	0.88	0.89	134
Cause-Effect(e2,e1)	0.87	0.89	0.88	194
Component-Whole(e1,e2)	0.80	0.75	0.78	162
Component-Whole(e2,e1)	0.66	0.65	0.65	150
Content-Container(e1,e2)	0.80	0.86	0.83	153
Content-Container(e2,e1)	0.84	0.67	0.74	39
Entity-Destination(e1,e2)	0.86	0.84	0.85	291
Entity-Destination(e2,e1)	0.00	0.00	0.00	1
Entity-Origin(e1,e2)	0.84	0.70	0.76	211
Entity-Origin(e2,e1)	0.95	0.77	0.85	47
Instrument-Agency(e1,e2)	0.43	0.27	0.33	22
Instrument-Agency(e2,e1)	0.69	0.63	0.66	134
Member-Collection(e1,e2)	0.64	0.72	0.68	32
Member-Collection(e2,e1)	0.79	0.83	0.81	201
Message-Topic(e1,e2)	0.78	0.73	0.75	210
Message-Topic(e2,e1)	0.83	0.49	0.62	51
Other	0.43	0.57	0.49	454
Product-Producer(e1,e2)	0.77	0.67	0.72	108
Product-Producer(e2,e1)	0.67	0.57	0.62	123
accuracy			0.72	2717
macro avg	0.71	0.66	0.68	2717
weighted avg	0.74	0.72	0.72	2717

TABLE IV
PERFORMANCE METRICS FOR BILSTM WITH GLOVE EMBEDDINGS AND GLOBAL MAXPOOLING

1) Embedding Representation: We used the power of word embeddings to improve the model's ability to understand the semantic relation between the words. Specifically, we have use pre-trained GloVe embeddings to capture the global Semantic information. Additionally, to include the contextual information, we also used TinyBert embeddings which is a lightweight variation of Bert. In contrast with the previous approach, neural networks really take advantage of the context given by these embeddings, even over SVMs. However, the computational cost is heightened by the need to train the neural network, and the increased complexity of the model may reduce its interpretability compared to simpler machine learning approaches.

2) BiLSTM: The BiLSTM architecture consists of two LSTM neural networks that process the input sequence in both forward and backward directions. This helps to effectively capture dependencies and the semantic relationships between entities. In our implementation, the input of the BiLSTM is obtained by embedding the text from the SemEval dataset capturing the contextualized information at each step taken. We have also tried applying regularization to the Recurrent Layers but found no noticeable improvements. For the feature extraction layer we have tested several approaches including attention, convolutional layers, *global*, *average* and *max* pooling, and found the latter to be the most effective. Furthermore, we enhance the performance by incorporating a **MaxPooling layer** to down-sample the input between hidden layers while keeping important features from fading away (Peng Zhou et al.). Finally, we add a fully connected layer followed by a *softmax* activation function. The fully connected layer takes the pooled representations and learns to classify the relationship between the entities based on this down-sampled contextual information. The softmax activation function produces a probability distribution over the possible relation classes enabling our model to learn and make accurate predictions on the SemEval 2010 Task 8 dataset.

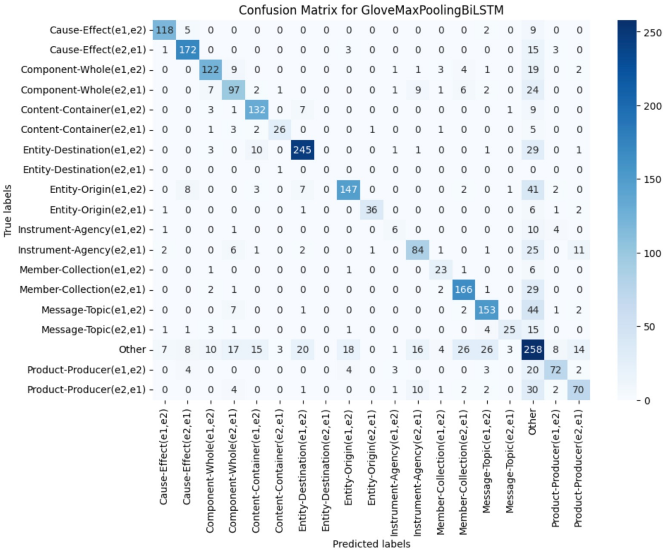

Fig. 3. Confusion Matrix for GloveMaxPoolingBiLSTM

IV. Discussion

Our report focuses on the strengths and weakness of different RE approaches and guiding the future research in the field of Natural Language Processing.

The traditional machine learning approaches like the Support Vector Machines had a very good performance in the SemEval 2010 Task 8 dataset. However, BERT embeddings with Naive Bayes negatively affected the model, which shows the challenges of integrating it with algorithms which assume feature independence since more steps would need to be taken to carry out the context given by the embeddings.

The deep learning technique, such as the employed BiLSTM network, demonstrated excellent results in multiple relationship types, proving its ability to effectively capture sequential dependencies and contextual data, improving relation extraction. Furthermore, the use of pre-trained embeddings, including GloVe and BERT, significantly contributed to the semantic comprehension of the text.

The final findings of our research indicate that selecting the most suitable method depends on the task and the characteristics of the dataset. While traditional techniques are straightforward, they can still provide excellent outcomes and are very interpretable. On the other hand, deep learning models, particularly those utilizing pre-trained embeddings, demonstrate superior performance in identifying linguistic features. However, this comes with the cost of reducing interpretability and increasing computational power.

V. Future Work

Moving forward, future research could focus on fine-tuning BERT embeddings for better integration with Traditional Machine Learning algorithms and investigate other advanced transformers. When it comes to Support Vector Machines, dimensionality reduction methods (PCA or t-SNE) have the potential to decrease training times while maintaining the non-linear *rbf* kernel. Moreover, incorporating additional pre-processing steps, such as Part-of-Speech (POS) tagging, Named Entity Recognition (NER), and dependency parsing, could provide the models with more informative features, leading to enhanced performance.

For BiLSTMs, future work could involve the implementation of sequence-to-sequence architectures, which have shown great promise in various Natural Language Processing tasks. Furthermore residual connections between the input and output of the BiLSTM layers could help in mitigating the vanishing gradient problem and allow for the training of deeper networks. Ultimately, employing a transformer-based encoder-decoder architecture in conjunction with the BiLSTMs could help capture long-range dependencies and improve the overall performance of the model.

In summary, for all the methods used, the use of additional linguistic features, such as POS tags and named entities, along with more advanced pre-processing techniques, should enhance the models' predictions. Fine-tuning the models on larger datasets and increasing the diversity of the training data could also lead to better generalization and overall performance.

References

[1] Hendrickx, Iris, Kim, Su Nam, Kozareva, Zornitsa, Nakov, Preslav, Ó Séaghdha, Diarmuid, Padó, Sebastian, Pennacchiotti, Marco, Ro-

mano, Lorenza and Szpakowicz, Stan. ”SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals”. Proceedings of the 5th International Workshop on Semantic Evaluation, July 2010, Uppsala, Sweden. Association for Computational Linguistics, pages 33-38. https://www.aclweb.org/anthology/S10-1006

[2] Hong, G. (2005). Relation Extraction Using Support Vector Machine. In Second International Joint Conference on Natural Language Processing: Full Papers.

[3] Zhang, S., Zheng, D., Hu, X., & Yang, M. (2015). Bidirectional long short-term memory networks for relation classification. In Proceedings of the 29th Pacific Asia conference on language, information and computation (pp. 73-78).

[4] Thillaisundaram, A., & Togia, T. (2019). Biomedical relation extraction with pre-trained language representations and minimal task-specific architecture. In Proceedings of the 5th Workshop on BioNLP Open Shared Tasks (pp. 84-89).

[5] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz and Polosukhin, Illia. ”Attention is all you need”. Advances in neural information processing systems, pages 5998-6008, 2017.

[6] Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton and Toutanova, Kristina. ”BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. CoRR, abs/1810.04805, 2018. http://arxiv.org/abs/1810.04805

[7] Vikramkumar, Vijaykumar B and Trilochan. ”Bayes and Naive Bayes Classifier”. CoRR, abs/1404.0933, 2014. http://arxiv.org/abs/1404.0933

[8] Zhou, Peng, Qi, Zhenyu, Zheng, Suncong, Xu, Jiaming, Bao, Hongyun and Xu, Bo. ”Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling”. CoRR, abs/1611.06639, 2016. http://arxiv.org/abs/1611.06639

[9] Zhang, Dongxu and Wang, Dong. ”Relation Classification via Recurrent Neural Network”. CoRR, abs/1508.01006, 2015. http://arxiv.org/abs/1508.01006

[10] Zhou, GuoDong, Su, Jian, Zhang, Jie and Zhang, Min. ”Exploring Various Knowledge in Relation Extraction”. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05), June 2005, Ann Arbor, Michigan. Association for Computational Linguistics, pages 427-434. https://aclanthology.org/P05-1053. doi: 10.3115/1219840.1219893