Navigating SemEval: A Comparative Study of Feature Engineering in SVM and BERT for Relation Extraction

Chandni Dewani[1], Rahul Sen[1], Ifra Moin[1], Abhinandan Das[1]
[1]Department of Social Science, University of Manchester, United Kingdom

Abstract

Relation Extraction stands as a crucial component of Natural Language Processing (NLP), focusing on deciphering meaningful associations between entities within textual data. This paper investigates two approaches for relation extraction on the SemEval 2010 Task 8 dataset: Support Vector Machines (SVM) and the BERT-base pre-trained transformer model. We extract a comprehensive feature set encompassing entity mentions, dependency information, and lexical knowledge from WordNet. The entity mentions are mapped to their closest equivalents within WordNet to enhance semantic consistency. Sentences are pre-processed to include only relevant terms and augmented with contextual definitions. We evaluate the performance of both models using the same feature set, enabling a direct comparison of their effectiveness in relation extraction on this benchmark dataset.

1 Introduction

Relation Extraction (RE) is a fundamental task to uncover meaningful connections between entities in a given text, carrying profound implications in downstream tasks such as question answering, knowledge graph construction, information retrieval, and sentiment analysis.

Recent studies indicate a prevalent inclination toward Neural Network models [2][9][11] for relation extraction tasks. These models leverage sentence encoding techniques to comprehend contextual information, assuming that each word contributes to relation classification. While these approaches often achieve good performance, this approach can introduce noise by assigning undue weight to irrelevant words. Additionally, alternative studies [7][5] that utilize features derived from lexical resources like WordNet or NLP tools such as dependency parsers and named entity recognizers (NER) can be limited by a predefined set of features, potentially neglecting rich contextual information.

This paper proposes a novel approach for relation extraction that addresses the limitations of existing methods by incorporating feature engineering and contextual enrichment techniques. The core of our proposed approach involves two key aspects:

1.1 Rich Contextual Feature Engineering: This process involves detailed entity analysis (including part-of-speech tags, dependency parsing, and surrounding words), focused sentence representation (extracting the relevant section containing both entities and filtering for specific word types), and custom entity disambiguation (using synsets, hypernyms, and cosine similarity to capture the contextual meaning of entities).

1.2 Enhanced Text Embedding: We utilize the Word2Vec module with a comprehensive series of checks. These checks attempt to find the best possible embedding representation for each word, including capitalized variations, hyphen removal, base form conversion, multi-word handling, rule-based adjustments, and segmentation as a last resort.

We investigate the effectiveness of two distinct

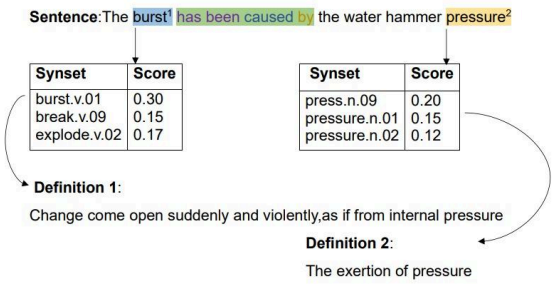

Figure 1. This figure depicts tailoring a sentence to focus on the relation between entities (superscripted ¹ & ²). Text colors highlight parts of speech (POS) used: purple for verbs, blue for auxiliary verbs and orange for adjectives.

learning models namely SVM, and BERT using our approach on the SemEval-2010 Task 8 dataset, which focuses on multi-way classification of semantic relations between entity pairs.

2 Related Work

Traditional machine learning models like Support Vector Machines (SVMs) have established themselves as a reliable choice for relation extraction due to their efficiency and effectiveness.[8] proposed a kernel-based SVM approach that utilizes various linguistic features, while [6] explores a cascade SVM framework for relation extraction.

In recent years, pre-trained language models like BERT (Bidirectional Encoder Representations from Transformers) have revolutionized the field of relation extraction. [1] demonstrated the effectiveness of BERT for relation extraction by fine-tuning the pre-trained model on labeled data for specific relation types. [4] further explored this potential by proposing a SpanBERT-based model for joint entity and relation extraction (ERE).

While both feature engineering and pre-trained language models have proven successful for relation extraction, there is a growing interest in combining their strengths. [10] suggested that this combination can lead to improved performance compared to relying solely on one approach. While traditional feature engineering for relation extraction often leverages lexical features [3], dependency parsing [8], and basic WordNet similarity [6], our approach builds on the emerging trend and enhances these by incorporating custom entity disambiguation and custom sentence representation. Specifically, we focus on capturing the fine-grained relationships between entities by including dependency features for both entities, and enriching the context by concatenating the contextual definitions retrieved from WordNet for each entity within the refined sentence.

3 Proposed Methodology

We opted for a two-pronged approach to explore the strengths of both feature engineering and contextual learning namely, SVM and BERT.

3.1 Preprocessing & Feature Engineering:

Our feature engineering process incorporates two key novelties: 1. We employ a custom entity disambiguation technique that determines the most relevant synset definition for each entity 2. The second novelty lies in our custom sentence representation technique.

A. Custom Entity Disambiguation: We extract entities from each sentence, creating separate columns for each, followed by extraction of their pos_tags, dependency tokens and tokens before and after the entities.We then capture the contextual meaning of entities using synset & hypernym extraction along with the definitions for the extracted synsets. We calculate the cosine similarity between the main words (nouns, verbs, adjectives) in the definition and the sentence (excluding the entity words) to determine the most relevant synset definition for each entity (Figure 1).

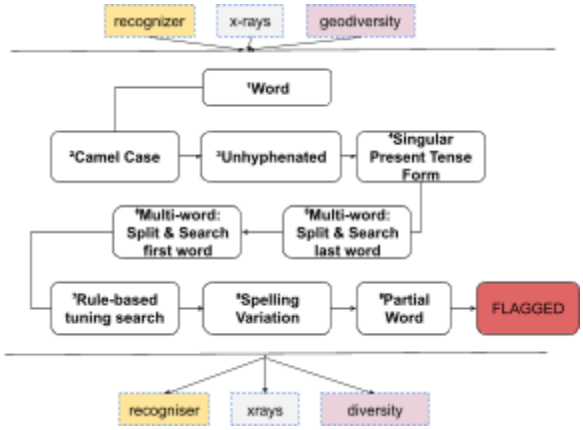

Figure 2.Word2Vec process with embedding representation checks.

Once both entity synsets are identified, we extract their lowest common hypernym to understand the inherent relationship between the entities.

B. Custom Sentence Representation: We tailor the sentence to focus on the relation between entities by selecting only the portion of the sentence between the entity occurrences as shown in Figure 1. Next, we trim the sentence by keeping only tokens with verb, auxiliary verb, and adjective POS tags between the entities. Finally, there are two variations of the trimmed sentence, one contains the trimmed sentence only and the other contains trimmed sentences concatenated with the definitions of both entities (derived from A) at the end of the sentence.

Both the chosen models take numerical inputs required to represent each data instance as a vector. For the custom sentences (Section B), we leverage the power of a pre-trained sentence transformer model called "all-MiniLM-L6-v2" which is a more lightweight and faster version of BERT capturing the core idea of contextual word embeddings. The remaining features are embedded using Word2Vec which assigns a unique vector to each word based on its usage within a large corpus. We implement a series of checks to ensure robust embedding representation as illustrated in Figure 2.

To ensure robust feature representation, we filter the training dataset based on the following criteria: if the vector representation for any feature is not present in the Word2Vec model (for SVM), or if a synset of any entity is not found (for SVM & BERT), the corresponding sentence is removed. We removed a total of 25 sentences for SVM model and 3 for BERT model from the training dataset, eliminating data points that might introduce noise or errors during model training.

3.3 SVM Approach

Our first approach focuses on implementation of SVM due to its established efficiency and effectiveness in relation extraction tasks. In this, we leverage the above described meticulously crafted comprehensive feature set to capture various linguistic properties relevant to relation identification.

3.4 BERT Approach

The second approach involves using the pre-trained BERT-base uncased model due to its excellent architecture aimed at capturing contextual information within sentences, making it a strong candidate for this task. Our model architecture employs BERT and incorporates relevant syntactic information (refined sentence and entities) within the tokenization process. Inputs are tokenized using the BertTokenizer and fed into a BERT-base-uncased model to generate contextual embeddings for each token. A pooled output representation is extracted from BERT's final layer and processed through a single fully-connected layer with dropout for regularization, ultimately classifying the relationship between entities. The model is trained using cross-entropy loss and the AdamW optimizer, with a validation set to monitor performance.

Train & Test Batch Size	32
Number of epochs	5
Dropout Rate	0.05
Learning rate	2e-5

Table 1. Hyperparameter Settings for BERT Model.

Features	F1-score
Refined Sentence + Entity	68
Refined Sentence + Entity + Dependency Tokens	73
Refined Sentence + Entity + Dependency Token + Previous & Post Tokens + Hypernym + Entity Definition	**74**
All features	74
Best Feature Set (Row 3) excluding 'Other' Class	**86**

Table 2. Experiments with feature sets used in SVM models on validation sets.

Table 1. shows the hyperparameters set in our proposed model. Furthermore, the parameters of the pre-trained BERT model are initialized according to [1].

4 Experiments & Results

4.1 Dataset & Evaluation Metric

To evaluate the performance of our approaches, we use the f1-score on SemEval 2010 task 8 dataset consisting of a total of 19 relations with nine directed relations. The dataset is partitioned into 8000 and 2717 instances for training & testing respectively.

4.2 Results

SVM Model: Table 2 explores the impact of different feature sets on the performance of an SVM model for relation extraction. F1-score increases with more features, but plateaus with additional ones. We achieved a 76% F1-Score on the test set. Our analysis suggests the "Other" class likely contains random sentences with no clear patterns, making it difficult to predict and lowering overall accuracy. By excluding this class and focusing on the remaining classes with the best feature set, our model achieves a strong F1-score of 86%. This indicates the model can effectively classify the well-defined categories.

BERT Model: We then investigated if leveraging a pre-trained language model like BERT could further improve performance with the best feature set obtained previously. An overall F1-score of 78% and 77% on validation & test sets respectively was achieved using the BERT-based approach.

5 Conclusion

In conclusion, our evaluation yielded comparable F1-scores between the feature-engineered SVM and the pre-trained BERT model, with BERT exhibiting a slight advantage. This highlights the effectiveness of well-designed features with its strengths in handling complex contexts without extensive feature engineering. While our evaluation showed comparable F1-scores between the feature-engineered SVM and the pre-trained BERT model, BERT achieved a significant 4% increase in results. This demonstrates BERT's capability to outperform traditional machine learning approaches on this task. SVM relies on well-designed features, offering interpretability and efficiency. However, feature selection complexity can limit its performance. BERT excels in handling intricate contexts without extensive feature engineering. Additionally, it scales well for large datasets. However, BERT's interpretability remains a challenge, and training it can be computationally expensive.

REFERENCES

[1] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[2] dos Santos, C., Nguyen, M., Pham, T., & Manning, C. D. (2015). Identifying semantic relations between named entities using convolutional neural networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 1414-1423). Association for Computational Linguistics.

[3] Gupta, A., Harrison, P.J., Wieslander, H., Pielawski, N., Kartasalo, K., Partel, G., Solorzano, L., Suveer, A., Klemm, A.H., Spjuth, O., Sintorn, I.-M. and Wählby, C. (2019), Deep Learning in Image Cytometry: A Review. Cytometry, 95: 366-380. https://doi.org/10.1002/cyto.a.23701

[4] Joshi, M., Chen, D., Liu, Y., Jiang, J., Xu, X., & Luo, X. (2020). SpanBERT for joint entity and relation extraction. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 6044-6057). Association for Computational Linguistics.

[5] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP), pages 1532– 1543

[6] Peng, H., He, S., & Zhao, J. (2014). Two-stage relation classification with cascade SVM. Information Sciences , 282, 350-360.

[7] Shen, T., & Huang, X. (2016. Feature engineering for relation extraction. Knowledge and Information Systems , 48(3), 872-894.

[8] Sun, A., Qiu, X., & Yu, Y. (2011). Exploiting multiple sources of knowledge for relation extraction. Proceedings of the 15th conference on computational natural language learning (pp. 1043-1052). Association for Computational Linguistics.

[9] Tianyu, Z., Xiaolong, Z., Jun, Y., & Yuming, L. (2019). A deep learning architecture for relation extraction. Neurocomputing , 370, 224-231.

[10] Wu, S., Sun, Y., & Li, X. (2023). A hybrid CNN-BERT approach for relation extraction. Knowledge and Information Systems (pp. 1-22). Springer.

[11] Zhang, Y., Sun, H., Meng, Z., Tang, J., & Luo, Z. (2022). A survey on neural relation extraction.