Relation Extraction (RE) for the SemEval 2010 Task 8

Yu Guo, Yunfang Hou,
Xinhong Yu, Shuainan Chen
School of Computer Science and Engineering, The University of Manchester, Great Manchester M13 9PL, England

Abstract

This study focuses on Relation Extraction (RE) within Natural Language Processing (NLP) using two approaches: traditional machine learning with Support Vector Machines (SVM) and a symbolic or unsupervised method, applied to the SemEval 2010 Task 8 dataset. The SVM approach leverages its strengths in high-dimensional spaces and nonlinear problem-solving, incorporating linguistic features to enhance model performance. In contrast, the symbolic method utilizes verb and preposition statistics for relation inference, emphasizing computational efficiency despite lower accuracy. Evaluation highlights the SVM's superior performance, indicating the complexity of RE and the need for extensive training data. This work contributes to RE research by demonstrating the effectiveness of integrating diverse techniques to address semantic relation challenges, offering insights for future studies.

1 Introduction

In the realms of knowledge graph construction, question-answering system enhancement, and the improvement of machine reading comprehension capabilities, Relation Extraction (RE) has emerged as a pivotal task within Natural Language Processing (NLP), with its significance increasingly acknowledged. In this coursework, we will employ two approaches: a traditional machine learning approach and a symbolic or unsupervised approach, to extract the relationship between two phrases in a sentence.

This paper explores the performance of RE in the face of abstract semantic relationship challenges by applying these two methods to the SemEval 2010 Task 8 dataset. Through comparison and analysis, we provide valuable methodological guidance for future RE research and applications, demonstrating the effectiveness of integrating diverse technologies to tackle specific domain RE challenges.

The SemEval-2010 Task 8 dataset focuses on multi-way classification of semantic relations between pairs of nominals. This task is designed to compare different approaches to semantic relation classification and serves as a standard testbed for future research. The dataset features various relations such as Cause-Effect, Component-Whole, Content-Container, Entity-Destination, Entity-Origin, Instrument-Agency, Member-Collection, Message-Topic, Product-Producer, and others, totaling 19 distinct relation types including an "Other" category for relations that don't fit into the predefined ones. The dataset comprises 8,000 training examples and 2,717 test examples, with the text data represented as strings and the relations as classification labels. It aims to provide a comprehensive framework for evaluating the performance of RE models across a diverse set of semantic relation types, which is critical for enhancing the models' generalization capabilities in real-world NLP applications.

2 Review

In our NLP group task on Relation Extraction (RE), we primarily focused on employing two methodologies for training our corpus: the Traditional machine learning-based approach and the Symbolic or unsupervised approach. Below, we will review these two approaches separately.

2.1 Traditional machine learning

The following are three research papers based on traditional machine learning.

Relation extraction using support vector machine[1] contributes new knowledge in the field of relation extraction by proposing a supervised machine learning approach using Support Vector Machines (SVM) for detecting and classifying relations in the Automatic Content Extraction (ACE) corpus. It introduces a two-staged extraction approach, dividing the task into relation detection and classification, and applies distinct linguistic features, including lexical tokens, syntactic structures, and semantic entity types, as well as the distance between

entities to improve performance. Additionally, the paper successfully utilizes entity distance as a feature for both relation detection and classification, and evaluates the system's performance in terms of recall, precision, and F-measure. Furthermore, the paper highlights the potential extension of the research to include semantic information and integrate entity recognition with relation extraction, providing valuable insights for future research in the field.

Exploring various knowledge in relation extraction[2] contributes new knowledge in the field of relation extraction by exploring the incorporation of diverse lexical, syntactic, and semantic knowledge in feature-based relation extraction using Support Vector Machines (SVM). It demonstrates the effectiveness of base phrase chunking information and the use of semantic information to improve performance, as well as outperforming previously reported systems on the ACE corpus. The study also illustrates the limited contribution of additional full parsing information and the potential for further enhancement through the incorporation of semantic resources such as WordNet and Name List. Additionally, the paper provides insights into the performance contributions of different features and the challenges associated with long-distance relations in relation extraction tasks.

UTD: Classifying Semantic Relations by Combining Lexical and Semantic Resources[3] contributes new knowledge in the field of semantic relation classification by presenting a system that combines lexical and semantic resources to achieve high accuracy in identifying semantic relations between nominals. The authors introduce a classification approach that first determines the type of semantic relation and then predicts the direction of the relation using SVM classifiers and a diverse set of features. This novel approach, along with the use of various linguistic resources, demonstrates the effectiveness of leveraging different types of information to enhance the identification of semantic relations in text. Additionally, the paper provides insights into the importance of background knowledge, context, and linguistic features in accurately classifying semantic relations, showcasing the significance of incorporating diverse resources for improved performance in semantic relation tasks.

2.2 Symbolic approach

Due to the complexity of abstract relation extraction, the traditional rule-based approach is not common in recent research. According to a paper[4] which reviews SemEval-2010 Task 8, only one of eleven participants used Decision Rules/Trees to classify the relation. This participant[5] used the Part-of-Speech tag to select 19 features and characteristics in the sentence to estimate the semantic relation. C4.5 decision trees and Cohen 1995's RIPPER algorithm were used in the prediction process. An accuracy of 26.67% was achieved with the testing set.

3 RE Approach Description

3.1 Traditional machine learning

We apply Support Vector Machines partly because it represents the state of the art performance for many classification tasks. The application of SVM in relation extraction tasks benefits from its performance in high-dimensional spaces, strong generalization ability, capacity to handle nonlinear problems, efficiency in processing sparse data, and customizability. These characteristics make SVM a powerful tool for executing complex NLP tasks like relation extraction. This learning algorithm has a robust rationale for avoiding overfitting.

After reading three classic papers on relation extraction implemented with traditional machine learning methods, I've adopted their experience in optimizing models. Notably, the correlation between the proximity of entities and the strength of their relational ties suggests a diminution in connection with increasing distance, thereby influencing our feature engineering strategy[1]. The integration of entity distance into our model aims to rectify the potential degradation in performance due to overly distant entity pairs. Moreover, the adoption of CountVectorizer facilitates the transformation of textual data into a term frequency matrix, incorporating unigrams, bigrams, and trigrams, as specified by the ngram_range parameter. This comprehensive approach, complemented by the extraction of syntactic and semantic attributes via the spaCy package, is poised to enhance the efficacy of our SVM-based model significantly.

The methodology for extracting relations can be succinctly outlined as follows:

1. Data acquisition involves loading from designated training and testing files.
2. The preprocessing phase entails the removal of specific markup tags and normalization of the textual data, including lowercase conversion and the elimination of irrelevant characters and terms.
3. Subsequent to preprocessing, the application of the spaCy package enables the extraction of parts of speech, tags, and dependencies of tokens, enriching the feature set for model training.
4. The CountVectorizer, followed by the application of the TfidfTransformer, constitutes our text transformation strategy, translating the corpus into a matrix representation that captures term frequency and the significance of terms within documents.
5. A pipeline is constructed to amalgamate all feature transformation processes, facilitating seamless integration and processing.
6. Model evaluation is conducted to assess the performance across various metrics.
7. Finally, the model is applied to new sentences to demonstrate its capability in identifying and classifying relations between nominated entities.

This academic investigation into the application of SVM for RE, grounded in the insights gained from a thorough review of existing literature, aims to contribute to the advancement of knowledge in the field and serve as a foundation for further research endeavors.

3.2 Symbolic approach

The second approach chosen is a symbolic approach without using machine learning. The main reason for choosing it is that a traditional linear program can save many computing resources, especially for those users who do not have a computing device with a powerful CPU and GPU. Also, the symbolic approach can be regarded as a reference to be compared with the machine learning approach, which is chosen as the second approach. Also, the suitability of Bootstrapping method has been considered, which leads to a negative conclusion. Because the relations (such as Cause-Effect and Component-Whole) in our chosen dataset are abstract, thus it is hard to find more corpus with labelled relations for bootstrapping.

The main method is to use the statistics of the VERB and ADP related to the entities in the sentence to infer the relation between them. It is based on an idea: the amount of commonly used nouns is ten times more than that of commonly used verbs. The action that can be done by entities is limited, but the very entities can be largely variable in different contexts. Also, the relation between two entities is largely determined by the verb. Python library spaCy is used for NLP manipulation in the following described approach. During the relation extraction process, the clause including the entities is separated to eliminate the interference from unrelated clauses. Then each token in the sentence is tagged with POS. The tokens with VERB or ADP tag are then extracted. The two tokens (one VERB and one ADP) with the lowest distance from any of the two entities are finally stored in a database with respective relations.

In the prediction process, the VERB and ADP tokens from the user input sentence will be extracted in the same way. The token will be compared with the database to find the relation with the highest possibility.

4 Evaluation Approach

4.1 Traditional machine learning

After reading the three papers listed above, we gained experience from their evaluation methods. To evaluate the model, we compute the recall, precision, and F1 score for each relation, enabling us to see the detailed results. Moreover, we calculate the overall accuracy, macro-average recall, and macro-average precision of all predictions to assess the performance across the entire test dataset. In addition, when running the notebook, users can also input sentences with tags (<e1></e1><e2></e2>) to test the model's performance.

For the test set in this dataset, the model's accuracy reached 60.4%, Macro_Average_Recall reached 54.1%, and Macro_Average_Precision reached 59.0%. Additionally, the model's F1-score for the relation "other" was the lowest, at

only 32.6%, while the F1-score for the relation "Entity-Destination (e1, e2)" was the highest, reaching 78.4%.

4.2 Symbolic approach

As stated before, the chosen dataset contains a training set and a testing set. The training set was used in the database generation process described in the previous section. After that, the testing set was inputted into the program to compare the predicted result. Correct prediction percentage, i.e. accuracy, is used as the evaluation metrics.

For the chosen dataset in this project, the program outputs 584 correct predictions out of 2717 testing sentences. The accuracy is 21%.

5 Discussion

Through comparison of prediction results, the accuracy of the symbolic approach is 21%, while the SVM-based RE model exhibits a higher accuracy of 60.4%, indicating a significant gap between the two. When running the code for both methods in a Colab T4 GPU environment, the SVM-based RE method required about 5 minutes, whereas the symbolic approach took approximately 40 minutes with mutliprocessing. It is evident that, on the SemEval 2010 Task 8 dataset with 8000 data entries, SVM surpasses the symbolic approach in both accuracy and running speed.

This can largely be attributed to data engineering in the SVM method. During the data engineering phase, we extracted POS, Tag, and dependency for each sentence, and since SVM can effectively handle high-dimensional data, it is capable of fitting well and completing complex relation extraction tasks. Additionally, given the complexity of language, the test set is likely to contain sentences vastly different from the training set, and SVM's strong generalization ability thus provides it with an advantage over the symbolic approach.

Overall, the strengths of SVM include higher prediction accuracy; shorter model fitting and prediction time; and excellent generalization capability, allowing it to perform well with unseen data and minimize the risk of overfitting. Its drawbacks include poor model interpretability. The core principle of SVM is finding an optimal hyperplane in the feature space to differentiate between categories. This hyperplane is chosen to maximize the margin between different categories and the hyperplane, a decision process that is difficult to understand in language processing. Moreover, SVM models are sensitive to parameter selection, with different parameters yielding vastly different results.

The advantages of the symbolic approach include strong model interpretability, as its rule-based methods are generally easier to understand and explain, aiding analysis and debugging. Additionally, it requires no parameter tuning, thus no need for parameter adjustments when dealing with different datasets. However, the symbolic approach's drawbacks include poor prediction results and longer running times when dealing with datasets like SemEval 2010 Task 8, which has only 8000 entries and some data lacking verbs.

References

[1] G. Hong, "Relation extraction using support vector machine," in *Natural Language Processing–IJCNLP 2005: Second International Joint Conference, Jeju Island, Korea, October 11-13, 2005. Proceedings 2*, 2005, pp. 366-377: Springer.

[2] G. Zhou, J. Su, J. Zhang, and M. Zhang, "Exploring various knowledge in relation extraction," in *Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05)*, 2005, pp. 427-434.

[3] B. Rink and S. Harabagiu, "Utd: Classifying semantic relations by combining lexical and semantic resources," in *Proceedings of the 5th international workshop on semantic evaluation*, 2010, pp. 256-259.

[4] I. Hendrickx *et al.*, "Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals," 2019.

[5] F. Celli, "UNITN: Part-Of-Speech counting in relation extraction," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, 2010, pp. 198-201.