SemEval-2010 Task 8: Relation Extraction Models using Stacking Classifier and Deep Neural Networks

Abhishek K, Kashish K, Tanya C, Souvik M
Department of Computer Science
The University of Manchester, Manchester, UK

Abstract

Relation extraction is an essential preprocessing step in the field of Natural Language Processing. In this paper we propose two novel approaches including the Machine Learning method and Neural Network method. The model is trained on SemEval-2010 Task 8 dataset, following a sequence of preprocessing steps including data cleaning, tokenization, lemmatization, entity masking etc. The first model uses the Stacking Classifier Approach where SVC and RandomForest are trained as the base models with the former being used as the final training model, called **"SVM+RF-SVM"**. The second model combines a deep learning architecture with 1-D Convolutional networks, hidden layers of Bi-LSTM network, and multi-head attention mechanisms, termed **"BiLSTM-CNN-Multi-Head Attention"**. The models are evaluated by comparing the accuracy and the F1 Score of both the approaches. Experimental results on the SemEval-2010 dataset potentially offer novel methods for relation extraction tasks.

Keywords: Relation extraction, NLP, SVM, Random Forest, Multi-head Attention, CNN, Stacking Classifier, Deep Learning, BiLSTM, GloVe embedding, TF-IDF.

1. Introduction:

Relation Extraction (RE) extracts the relational concepts from the sentences provided in the form of plain text.

Sentence: "<e1>Chris</e1> is the president of the <e2>Student Union</e2>"		
Entity 1: *Chris*	**Entity 2:** *Student Union*	**Relation:** *Member-Collection (e1, e2)*

Table 1: Entity-Relation Structure

Here we focus on the semantic relationship between the two nominals e1 and e2 and extract the relation between them. It aids in highlighting the structured keywords for various downstream applications. Semantic relation extraction task is aiming to understand the relationships between entities in sentences. This paper presents a comprehensive approach utilising and modifying both traditional machine learning techniques and Neural Networks. The research demonstrates effectiveness of SVM and Random Forest stacking classifiers in capturing the relational patterns from each sentence. Furthermore, it explores the capabilities of Neural networks to automatically learn intricate hierarchical representations of semantic relations.

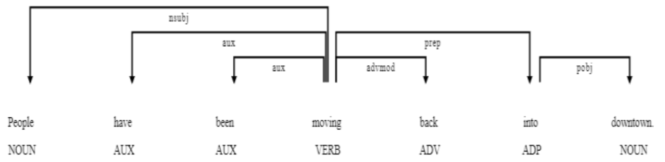

Figure 1: Dependency Graph

2. Related Work

In this section, we critically review three seminal papers that contribute significantly to the field of relation extraction using Machine Learning and deep neural networks (DNN).

Firstly, in the study by [5] **Iris et al (2010),** SVMs were utilised by the UTD system for semantic relation classification. It was implemented within a boosting-based classification strategy.

Another study done by [4] **Brian et al. (2010)** uses an approach classifying the semantic relation first, then using a relation-type-specific classifier to determine the relation's direction between pairs of nominals using SVM classifiers.

For Neural based approach, [1] **Zhang et al. (2015)** introduced bidirectional LSTM networks for relation classification using its capability to capture contextual information from both past and future sequences.

Another related work is the Attention-based BiLSTM model by [3] **Zhou et al. (2016).** This model focuses on relevant parts of input sequences, enhancing semantic relationship extraction.

Finally, our work is related to the Multi-level Attention model by [2] **Wang et al. (2016)** which presents an architecture combining CNNs with attention mechanisms to capture hierarchical features and contextual features of input data, improving classification.

Our approaches in Section 3 are integrating these related concepts and exploring additional features from NLP tools to build novel models for enhancing semantic relation extraction.

3. Methodology

3.1. Stacking Classifier - Using Support Vector Machine (SVM) and RandomForest

In this method, we use a Stacking Classifier, an ensemble method where the output from multiple classifiers also known as the base models is passed as an input to a final classifier model which uses this information to come up with the final classification. We use 2 classifiers, Support Vector Machine (SVM) and RandomForest as the base models and use Support Vector Machine (SVM) as our final model.

SVM is a supervised learning approach, classifying the specific type of relations on the train dataset by selecting a hyperplane that shows the maximum distance between two classes [7]. The second hybridized base model, Random Forest Classifier [8], is a supervised learning approach, which uses multiple decision trees and gives the output mode of the classes for classification [9].

SVC and Random Forest as base models [10] are suitable because of their complementary nature. SVC is then chosen as the final model due to its rationale for avoiding overfitting [11] and its ability to handle non-linear data by making use of the kernel functions [12].

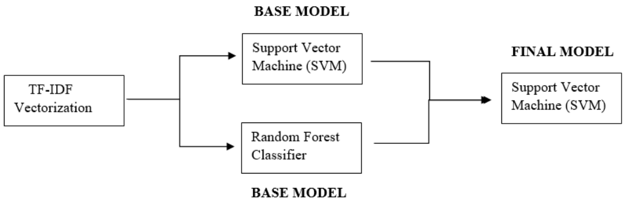

Figure 2: Stacking Classifier Flow Diagram

3.1.1 Features Used

In the exploration of Relation Extraction using the Support Vector Classifier approach, feature engineering and extraction act as the critical components. We start with entity extraction and identify the entities present in the sentence and their respective positions.

The entities present between the nominals indicate the relation present. For example, essays and volume indicate the Member – Collection relation. Following this, the data is normalised by removing the HTML tags, providing a uniform lowercase format, and eliminating the non – alphanumeric characters. This aids in drilling down the data to its semantic core. [13] Tokenization provides a list of words and punctuation called ‘Tokens’ at a more granular level which enables the model to consider the potential role of each word and define the corresponding relationship. With POS Tagging and Lemmatization, the tokens are assigned grammatical roles like Noun (NN), Verb (VBD) etc and then are reduced to their base forms which allows the model to focus on the intended meaning of the word rather than the different variations.

3.1.2. TF-IDF vectorization

It provides a numerical representation of the text present in the cleaned sentence which focuses on the entities that have greater emphasis on the relation context [14]. For instance, words like ‘cradle’ and ‘disassembler’ will show a higher TF-IDF score, showing their relational structure importance.

3.2 Neural Network Approach

In this section we propose Att-CNN-BiLSTM model with multi-head Attention mechanism. The model can be divided into components:

1. Input layer: It serves as the entry point, receiving input sentences for subsequent processing.
2. Embedding layer: It’s employed to transform each word within the sentences into low-dimensional vectors.
3. A combination of one BiLSTM layer and three Convolutional Neural Networks (CNN) is leveraged to extract high-level features from embedded representations.
4. Multi-Head Attention layer: It refines the feature extraction process by generating weight vectors.
5. Classifying: Utilising a softmax layer for relation classification, transforming the final vector into probability distribution over relation labels.
6. Optimizer: Nesterov Adam optimization

3.2.1. Word Embeddings

Given a sentence consisting of T words $S = \{x_1, x_2, . . . , x_T\}$, every word x_i is represented by its pre-trained [15] GloVe embedding ei of size d_{glove}. For each word xi, we check the pre-trained GloVe embedding from a pre-built matrix GloVeMatrix of size $|V| * d_{glove}$, where $|V|$ is the size of the GloVe vocabulary.

$e_i = GloveMatrix[x_i]$

Here, *GloveMatrix[x_i]* represents the GloVe embedding vector corresponding to the word x_i.

The sentence is then represented as a set of GloVe embedding $emb_s = \{e_1, e_2, ..., e_T\}$. This set of embedding can be fed into the next layer of your neural network for further processing. The size of the GloVe embedding, d_{glove}, is typically a hyperparameter that we can choose based on dimensionality of the GloVe vectors [16].

3.2.2 Bidirectional Network

The proposed neural network model represents a comprehensive approach to relation classification tasks. The initial embedding layer, initialised with pre-trained word embeddings using GloVe 100 dimensions and a dropout rate of 0.5, establishes a foundational layer [17] for the model by capturing semantic representation of the input sequences. Our model, as depicted in [6] Figure 3, integrates a bi-directional LSTM layer with 200 units which plays a crucial role in encoding sequential information [18], allowing the model to capture contextual nuances within the input sequences followed by three parallel convolutional layers (CNN) with distinct kernel sizes (k) of 3,4 and 5, each having a filter size 100 units [19]. The CNN layers enable the extraction of hierarchical features and patterns. The outputs from the three CNN blocks undergo global max-pooling and are then concatenated. Throughout the model, except for the final layer using softmax activation, relu activation is employed. To enhance l2 regularisation, dropout is applied to the LSTM layer and to each CNN block after global max-pooling, with a dropout of 0.3. The integration of a multi-head attention mechanism with 8 heads and a key dimension of 64 introduces vital elements for capturing parts of the sentence which are most influential with respect to the two entities of interest [2].

A multi-head attention allows the model to attend to different parts of the input sequence simultaneously. Each attention head learns distinct sets of attention weights, enabling the model to capture various aspects of the input's context and dependencies. This parallel processing of attention heads enables the model to capture both local and global patterns [20] within the data. Finally, the features encoded by the CNN layers are fed into multiple dense layers with 300 and 200 units, followed by a prediction layer. For training, we have used Nesterov Adam optimization and categorical cross-entropy loss with a learning rate of 0.002 [21]. The rationale behind this architectural design is that the sequential LSTM layer serves as a feature encoder capturing patterns from sequences composed of semantic word embeddings. The subsequent CNN layers then can encode the category related features introduced by the LSTM [22].

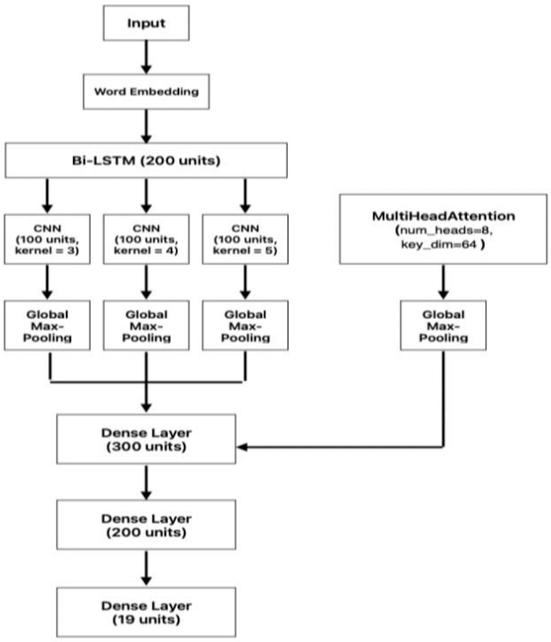

Figure 3: BiLSTM-CNN-Multi-Head Attention Architectural Diagram

3.2.3 Features Used

Our data preprocessing approach initiates with Data Cleaning which eliminates special characters and tags. Further, Padding and Tokenization convert words into numerical representations and enforce uniform sentence lengths. Subsequently, Entity Masking [23] highlights key entities <e1> and <e2> as '1' enabling focused attention on relevant features during the training stage. Next, GloVe Embedding further enhances the model's understanding of word context by mapping words to dense vectors based on semantic relationships. This entire pre-processing pipeline ensures that textual data is optimally prepared for subsequent model training, thereby enhancing the model's generalisation and performance for relation extraction tasks.

4. Results

The performance of both approaches is evaluated based on the F1 Score and accuracy metrics. The overall performance for both approaches is shown the Table-2.

Class	F1 Score - Stacking Classifier	F1 Score - BiLSTM-CNN-Multi-Head Attention	Support
Cause-Effect (e1, e2)	0.83	0.89	134
Cause-Effect (e2, e1)	0.81	0.87	194
Component-Whole(e1, e2)	0.50	0.79	162
Component-Whole (e2, e1)	0.42	0.72	150
Content-Container (e1, e2)	0.71	0.82	153
Content-Container (e2, e1)	0.70	0.74	39
Entity-Destination (e1, e2)	0.78	0.87	291
Entity-Destination (e2, e1)	0.00	0.00	1
Entity-Origin (e1, e2)	0.69	0.82	211
Entity-Origin (e2, e1)	0.71	0.86	47
Instrument-Agency (e1, e2)	0.40	0.52	22
Instrument-Agency (e2, e1)	0.55	0.66	134
Member-Collection (e1, e2)	0.29	0.62	32
Member-Collection (e2, e1)	0.61	0.87	201
Message-Topic (e1, e2)	0.61	0.86	210
Message-Topic (e2, e1)	0.52	0.78	51
Product-Producer (e1, e2)	0.47	0.74	108
Product-Producer (e2, e1)	0.37	0.70	123
Other	0.31	0.45	454
Overall	**0.58**	**0.75**	2717

Table 2: F1 score on Test dataset.

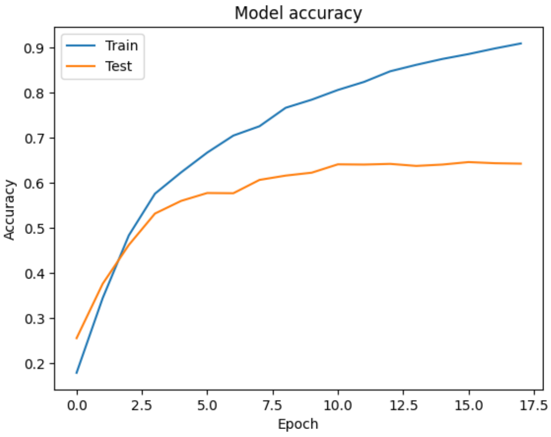

Figure 4: Train-Test Model Accuracy (BiLSTM-CNN-Multi-Head Attention)

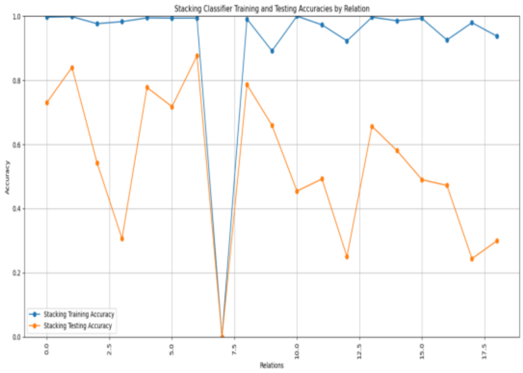

Figure 5: Train-Test Model Accuracy (Stacking Classifier)

5. Discussion and Conclusion

After evaluating the models, we observe that the Neural Network approach (NN) outperformed the Machine Learning approach (ML), achieving a 75% F1 score compared to ML approach's 58%. One prominent factor contributing to the superior performance is NN's ability to autonomously learn complex patterns and extract relevant features enabling it to capture intricate relationships effectively. The multi-head attention in the architecture allowed for hierarchical feature extraction, enhancing its capability to understand contextual meaning within input sentences. This comprehensive approach enabled the NN model to perform better for Relation extraction tasks.

On the other hand, despite the simplicity and interpretability of ensemble methods like stacking classifiers with SVM and RandomForest, the ML model struggled to handle complexity, intricate patterns and non-linear relationships inherent in relation extraction tasks. To handle the imbalance in the dataset and extract unbiased performance, we have used 'MICRO' averaged F1 score for evaluation. The frequency of relation 7(Entity-Destination) in the testing data is '1'. Due to this lack of data the model is not able to generalise well and consequently reducing the overall F1 score and accuracy of our models. Overall, while both approaches offer distinct advantages and limitations, comparatively Neural Networks handles complexity better than Machine Learning approach for relation extraction tasks.

6. ACKNOWLEDGEMENTS

The authors would like to thank netik1020 for providing base code and insightful comments via GitHub repository.

7. REFERENCES

[1] Zhang, S., Zheng, D., Hu, X., & Yang, M. (2015, October). ‘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).

[2] Wang, L., Cao, Z., De Melo, G., & Liu, Z. (2016, August). ‘Relation classification via multi-level attention cnns’, In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1298-1307).

[3] Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B. (2016, August). ‘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)* (pp. 207-212).

[4] Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Séaghdha, D. O., Padó, S., ... & Szpakowicz, S. (2019). ‘Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals’, *arXiv preprint arXiv:1911.10422*..

[5] Rink, B., & Harabagiu, S. (2010, July). ‘Utd: Classifying semantic relations by combining lexical and semantic resources’, In *Proceedings of the 5th international workshop on semantic evaluation* (pp. 256-259).

[6] Wiedemann, G., Ruppert, E., Jindal, R., & Biemann, C. (2018). ‘Transfer learning from lda to bilstm-cnn for offensive language detection in twitter’, *arXiv preprint arXiv:1811.02906*.

[7] Awad, M., Khanna, R., Awad, M., & Khanna, R. (2015). ‘Support vector machines for classification’, *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, pp. 39-66.

[8] Demidova, L. A., Klyueva, I. A., & Pylkin, A. N. (2019). ‘Hybrid approach to improving the results of the SVM classification using the random forest algorithm’, *Procedia Computer Science*, *150*, 455-461.

[9] Ramanathan, T. T., & Sharma, D. (2017). ‘Multiple classification using svm based multi knowledge-based system’, *Procedia computer science*, *115*, 307-311.

[10] Arnroth, L., & Fiddler Dennis, J. (2016). ‘Supervised Learning Techniques: A comparison of the Random Forest and the Support Vector Machine’.

[11] Lachaud, A., Adam, M., & Mišković, I. (2023). ‘Comparative study of random forest and support vector machine algorithms in mineral prospectivity mapping with limited training data’, *Minerals*, *13*(8), 1073.

[12] Shang, C., Huang, X., & You, F. (2017). ‘Data-driven robust optimization based on kernel learning’, *Computers & Chemical Engineering*, *106*, 464-479.

[13] Pivovarova, L., & Yangarber, R. (2018). ‘Comparison of representations of named entities for multi-label document classification with convolutional neural networks’, In *Proceedings of The Third Workshop on Representation Learning for NLP*. The Association for Computational Linguistics.

[14] Dalaorao, G. A., Sison, A. M., & Medina, R. P. (2019, October). ‘Integrating collocation as tf-idf enhancement to improve classification accuracy’, In *2019 IEEE 13th International Conference on Telecommunication Systems, Services, and Applications (TSSA)* (pp. 282-285). IEEE.

[15] Pennington, J., Socher, R., & Manning, C. D. (2014, October). ‘Glove: Global vectors for word representation’, In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

[16] Camacho-Collados, J., & Pilehvar, M. T. (2020, December). ‘Embeddings in natural language processing’, In *Proceedings of the 28th international conference on computational linguistics: tutorial abstracts* (pp. 10-15).

[17] Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., ... & Ward, R. (2016). ‘Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval’, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*(4), 694-707.

[18] Niu, Z., Zhou, M., Wang, L., Gao, X., & Hua, G. (2017). Hierarchical multimodal lstm for dense visual-semantic embedding. In *Proceedings of the IEEE international conference on computer vision* (pp. 1881-1889).

[19] Zhang, D., & Wang, D. (2015). ‘Relation classification via recurrent neural network’. *arXiv preprint arXiv:1508.01006*.

[20] Ahmad, W., Kazmi, B. M., & Ali, H. (2019, December). ‘Human activity recognition using multi-head CNN followed by LSTM’, In *2019 15th international conference on emerging technologies (ICET)* (pp. 1-6). IEEE.

[21] Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., ... & De Freitas, N. (2016). ‘Learning to learn by gradient descent by gradient descent’, *Advances in neural information processing systems*, *29*.

[22] Fan, C., Li, Y., & Zhu, J. (2023, August). ‘Research and Application Based on CNN-LSTM’, In *2023 IEEE International Conference on Sensors, Electronics and Computer Engineering (ICSECE)* (pp. 800-803). IEEE.

[23] Jiang, H., Cao, T., Li, Z., Luo, C., Tang, X., Yin, Q., ... & Yin, B. (2022). ‘Short Text Pre-training with Extended Token Classification for E-commerce Query Understanding’, *arXiv preprint arXiv:2210.03915*.