Comparative Analysis of SVM and CNN-Bi-LSTM Models for Relation Extraction from Text Data

Pol Rovira-Wilde, Marios Avraam, Rakshit Yadav, Ayush Naudiyal
University Of Manchester

Abstract

This paper explores the task of Relation Extraction (RE) using the SemEval-2010 Task-8 dataset. We explore traditional machine learning and deep learning approaches, with a focus on feature engineering and extraction, and the pivotal role of text preprocessing. We implemented two distinct models: a hybrid Convolutional Neural Network and Bidirectional Long Short-Term memory (CNN-Bi-LSTM) model leveraging GloVe embeddings, and a Support Vector Machine (SVM) model using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. We compare the efficacy of these models based on precision, recall and F1-score, finding that while both models demonstrate competitive performance, the SVM model with TF-IDF vectorization offers advantages in terms of simplicity, computational efficiency, and ease of implementation. The SVM model achieved a macro F1-score of 0.71, and the CNN-Bi-LSTM model achieved a score of 0.68.

1 Introduction

RE is the process of categorizing the semantic relationship that exists between entities within text data. An effective RE approach will capture the contextual information about the target entities from a sentence and use this information to categorise the entitys' relationship.

The importance of RE is underpinned by its synergy with various other fields within NLP, which can be performed with a greater level of granularity and efficacy when enhanced by RE. A recent example of this can be seen in Ojokoh et al's work, which demonstrated how pattern-based relation extraction methods could be used to increase the performance of question-answering systems (2023).

There are several approaches which can be used to perform RE, this paper will focus on traditional machine-learning and deep learning approaches. A major difference between these two approaches is that traditional machine-learning is reliant on handcrafted features, whilst deep learning approaches are less reliant on feature engineering as they have the capacity to learn informative features.

Given the nature of RE, it's heavily dependent on other text pre-processing steps, which means that errors within these previous tasks can have a profound impact on the final result. For this reason, this article will place a strong emphasis on the role of text pre-processing.

2 Dataset

The SemEval-2010 Task8 dataset, designed as a benchmark for relation extraction tasks, comprises a diverse web-sourced corpus, enhancing model generalizability and capturing real-world variations, including informal grammar structures (Kata Gabor, et al. 2018). Its manually annotated entities and relations ensure reliable labels, crucial for managing the dataset's inherent heterogeneity. The inclusion of the 'OTHER' category, constituting 17.6% of the training data (far more than any other category), introduces imbalance into the dataset (See appendix B), posing challenges for unbiased training and performance evaluation (Hendricx et al. 2010).

The SemEval-2010 Task-8 dataset strikes a balance between comprehensiveness and exclusivity by ensuring that each nominal pair belongs to a single relation category, accounting for the complexities of natural language. Importantly, it accounts for relation directionality by treating entity pairs in different orders as separate classes. For instance, the dataset includes both *Component-Whole(e1,e2)*, signifying that the first entity (e1) is a component of the second entity (e2), and *Component-Whole(e2,e1)*, indicating that the second entity (e2) is a component of the first entity (e1). This fact adds granularity to the relation extraction capabilities of our models.

3 Literature Review

3.1 SVM

SVM has been a popular choice for RE due to its robustness in handling high-dimensional data spaces. Numerous studies, including those by GuoDong et al. (2005), Rink & Harabagiu (2010), and Detroja et al. (2023), have demonstrated SVM's effectiveness in classifying complex relations using various feature sets and kernels. For instance, the ClaiRE system, designed for SemEval-2010 Task-7, employed SVM along with hand-crafted features and word embeddings (Hettinger et al., 2018). These systems often incorporate SVM in conjunction with a rich set of features, including lexical cues and semantic role associations, to achieve notable precision and recall in RE tasks. Moreover, Akbani et al. (2004) explored the combination of SVM and SMOTE on imbalanced datasets, highlighting that this synergistic approach can significantly improve performances on such datasets where class imbalances pose challenges.

Complementing the classification power of SVM, the Term Frequency-Inverse Document Frequency (TF-IDF) feature weighting method has proven influential in enhancing feature extraction for RE. Studies such as the novel text mining approach by Dadgar et al. (2016) which achieved high classification precision in news categorization, showcase TF-IDF's ability to highlight significant terms, thus improving the model's focus on relevant features for relation classification. Further expanding the application of TF-IDF, the study by Singhania et al. (2022) introduces a novel task aimed at forecasting document relevance for relation extraction. By incorporating TF-IDF-weighted bag-of-words and n-gram representations, the research reveals the capabilities of these features in predicting relation types.

3.2 CNN and Bi-LSTM

CNN models are renowned for their ability to extract important features whilst reducing dimensionality. Studies have shown that CNNs are proficient in text classification tasks, like RE. This was proven by Zeng et al's (2014) study that demonstrated how a CNN model was able to extract meaningful features at both a lexical and sentence level from unstructured text data, achieving a high level of performance in RE tasks without the need for extensive feature engineering. The capacity for feature learning shown by CNN in this paper inspired our decision to implement a convolutional layer into our deep learning model.

Zhang et al (2015) used a Bi-LSTM approach for RE on SemEval-2010 Task 8, which achieved a high level of performance (macro F1 score of 83.6). Traditional LSTM layers process sequential data in temporal order, which means that tokens are processed in the context of preceding tokens. As Zhang et al explain when justifying their approach, this can be problematic for RE as the cues that signal entity relationship types may occur anywhere in a sentence. A Bi-LSTM layer remedies this problem by processing tokens in both a forward and backward direction. Therefore, to ensure our model captures important information regardless of its position in the sentence, we decided to use a Bi-LSTM layer in our deep learning model.

4 Methodology

4.1 Approach 1: CNN - Bi-LSTM Model

In our first approach to extracting relation classes from the SemEval 2010 Task 8 dataset, our strategy centered on a hybrid CNN and Bi-LSTM model which utilised spaCy, RegEx and PySpellChecker for text pre-processing, and Global Vectors for Word Representation (GloVe) word embeddings.

The addition of spaCy provided tokenisation and lemmatisation, which was supplemented with RegEx patterns that targeted the unconventional use of parentheses. Many of the tokens in the dataset without corresponding GloVe embeddings were composed of two distinct words separated by a parenthesis rather than whitespace, likely due to mistyping. The RegEx we used ensured these patterns were separated into 3 distinct tokens, which decreased the number of tokens in our dataset that were not recognised in GloVe's vocabulary, thus maximising the utility of this feature.

GloVe word embeddings were chosen because of their renowned ability in identifying word analogies as a result of utilising global word co-occurrence statistics (Kulshretha and Lodha, 2023). This makes GloVe particularly adept for RE as two entity pairs of the same relationship class can be viewed as analogous, which our model can leverage to improve its performance. This idea is rooted in Rosiello et al's conception of RE as an analogy problem (2019). As the effectiveness of using these embeddings is heavily reliant on our tokens existing within GloVe's vocabulary, we used PySpellChecker to correct misspelled tokens and thus decrease the number of tokens without corresponding GloVe embeddings.

To preserve the pre-trained knowledge within GloVe's embeddings, we only **unfroze** the word embeddings for tokens **without corresponding GloVe embeddings**, thus allowing our model to learn them during training. This approach was superior in our testing in comparison to randomly generating word embeddings for these tokens. Whilst this strategy increased the potential for overfitting, given the small number of remaining tokens without corresponding GloVe embeddings and their low frequency, this drawback was minimal.

For our model architecture we adopted a hybrid approach that, after the embedding layer, used a convolutional layer to capture local features followed by a Bi-LSTM layer to capture long-range dependencies. Since convolutional layers are well suited to extracting local features such as n-grams, this would allow our model to incorporate information about patterns within each sentence that were indicative of entity relationships. As our dataset contains sentences of varying lengths, we used a wide range of filter sizes to extract local features from both short and long sequences.

Following our CNN layer, we used a max pooling layer to reduce the dimensionality of the dataset for subsequent layers. Whilst this may result in a loss of information as sequence lengths are compressed, potentially reducing the effectiveness of the subsequent Bi-LSTM layer, the reduction in dimensionality significantly reduced training times, which is important given that Bi-LSTMs are computationally intensive. Furthermore, the loss of information is partially offset by the pooling layer's ability to retain the most important features in the dataset, which means it can provide noise reduction as not all words contribute equally towards a sentences' meaning.

The Bi-directional LSTM layer is vital for relation extraction as the type of relationship between entities may rely on cues that precede or follow the entities themselves. Therefore, this layer enables our model to utilise more of the context of each sentence when classifying relationships.

The stacking of both CNN and Bi-LSTM layers provides our model with a greater learning capacity, however, this increase in model depth raises the number of learnable parameters, which elevates the risk of overfitting. Therefore, we implemented the following regularization techniques: 1) a drop out layer with an aggressive dropout rate of 0.6 to ensure that our model didn't become highly dependent on specific features, and 2) a weight decay of 1e-4 was added to our Adam optimiser to incentivize the model to choose smaller parameter weights, and thus reduce model complexity, during the optimisation process.

Lastly, to speed up convergence, a leaky ReLU activation function was used to minimize the optimization problems caused by unstable gradients during backpropagation.

4.2 Approach 2: SVM

In our second approach to relation extraction, we adopted TF-IDF for text vectorization and SVM for classification. We began by removing the single-instance class 7 (see appendix A) from both the training and test sets, preventing potential model bias and overfitting. Removing class 7 also allowed us to apply SMOTE to address the issue of class imbalance, as this oversampling method requires more than one instance to function properly (Chawla et al., 2002).

SMOTE is an oversampling technique that synthetically generates new instances of minority classes based on similarities in the feature space. This technique is particularly effective in creating a more balanced class distribution, thus reducing bias towards majority classes, and improving the model's ability to accurately learn and classify underrepresented classes. Applying SMOTE significantly enhanced the representation and F1-scores of minority classes like 10, 12, and 15 by 0.1, 0.07 and 0.2 respectively. This enabled the model to learn and classify minority classes better without notably impacting the performance on well-represented classes.

For preprocessing, we leveraged SpaCy, coupled with NLTK's access to WordNet and an extensive list of English words (Sawicki et al., 2023). SpaCy was used for lemmatization and part-of-speech tagging, which converted words into their base forms, while enriching them with syntactic information. This process improved the model's ability to learn from generalized word representations and understand contextual relationships.

Additionally, we employed SymSpell (rmdort, 2017), a spell correction library, targeting words not found in NLTK's word lists, utilising a caching mechanism for efficiency. Non-rectifiable words were marked with a unique identifier to indicate uncertainty. Moreover, we calculated the distance between entities in each sentence, a crucial feature for relation classification (Jin et al., 2020).

While entity markers were preserved in their original form, non-alphabetic characters were removed, focusing the model's attention on meaningful linguistic content. However, we decided to exclude dependency parsing from our preprocessing pipeline, as tests indicated that it reduced model performance.

Given our prior use of word embeddings in our deep learning model, we adopted TF-IDF vectorization as our second approach. Unlike neural networks, which can adapt to misspelled words (Hu et al., 2021), TF-IDF offers the flexibility to vectorize any term without relying on predefined dictionaries, making it well-suited for supervised tasks. Its efficiency in high-dimensional spaces and its de-emphasis of frequent, uninformative words (Luthfi Ramadhan, 2021), makes it a great option for text classification problems. TF-IDF evaluates word significance within documents relative to the whole corpus, thereby distinguishing word relevance based on context. Moreover, the sparse, high-dimensional matrices generated by TF-IDF pair well with SVM's capability at handling high-dimensional feature spaces (Das & Chakraborty, 2018). This synergy is particularly powerful, as TF-IDF's emphasis on term uniqueness often yields linearly separable data representations, ideal for SVMs.

The choice of TF-IDF was further reinforced using an n-gram range from 1 to 3, allowing the model to consider not only individual words, but also combinations of up to three consecutive terms (Qin et al., 2021). This approach was particularly beneficial for capturing contextual information and the intricacies of entity relationships within sentences, greatly enhancing the model's ability to discern and accurately classify relational types.

As mentioned above, to address the NLP task of relation extraction, we chose SVM for its proficiency in identifying optimal hyperplanes that effectively separate high-dimensional data. Specifically, we employed the linear kernel for the SVM model (Patle & Chouhan, 2013), hinting at the linear separability of classes after being projected in the transformed feature space.

The choice of SVM was also influenced by its ability to address class imbalances through the 'balanced' class_weight parameter setting (Jason Brownlee, 2020), which adjusts class weights inversely to their frequencies, ensuring fair representation across all relation types. Combined with SMOTE, this significantly improved the model's performance, allowing our SVM with a linear kernel to strike an optimal balance between simplicity and the ability to navigate complex, high-dimensional spaces effectively.

5 Evaluation

To evaluate the performance of the CNN-Bi-LSTM model with GloVe embeddings and the SVM model with TF-IDF vectorization, we analyzed precision, recall, and F1-score. Precision measures the proportion of correct positive predictions out of all positive predictions, while recall measures the proportion of correct positive predictions out of all actual positive instances in the data. The F1-score, being the harmonic mean of precision and recall, provides a balanced assessment of a model's predictive abilities. These metrics offer insights into the strengths and weaknesses of the models in accurately predicting each relation type. For a comprehensive assessment, we employed the macro F1-score, which accounts for the class imbalance present in our dataset. This ensures each class contributes equally to the overall performance metric. Our evaluation approach aligns with the standard practices established for the SemEval-2010 Task8 (Zhao et al., 2021), enabling direct comparison with other research efforts in the field of relation extraction. In our approach, we used a held-out validation set to avoid overfitting, while also fine-tuning our hyperparameters, to ensure the best overall performance within our computational constraints.

The SVM model, with its linear kernel, outperformed in precision for classes 12 and 14 (see appendix C), whereas the Bi-LSTM showed strength in class 10. Both models followed a similar recall trend (see appendix D), suggesting the dataset's class structure generally influences class detection capabilities. The SVM model slightly edged out in F1-scores, particularly due to its precision, leading to a macro F1-score of 0.71 against CNN-Bi-LSTM's 0.68 (see Figure 1) for individual class performance). This suggests that SVM's robustness across various classes and computational efficiency gave it an edge, while also indicating that the complexity of neural networks may not always translate to superior performance.

A notable challenge for both models was predicting instances of class 10 and 18, due to less distinct patterns and class 10's limited examples. Class 18 (Other) posed inherent difficulties due to its diverse nature of entity types.

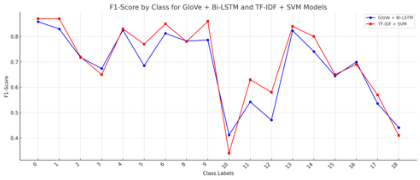

Figure 1: F1 scores by class

Despite these challenges, SVM showed better performance in 13 out of 18 classes, highlighting the benefits of utilizing n-grams (1,3) and SMOTE for contextual understanding and class balance (see appendix E). However, it should be noted that the increased dimensionality from implementing these two methods presents a trade-off with computational overhead.

One potential cause of the CNN-Bi-LSTM model's poorer performance may be due to the order of the layers used. The use of convolutional and pooling layers results in information compression, where contextual and sequential information necessary for understanding entity relationships over longer text spans, may be lost. This may have inhibited the following Bi-LSTM layer's capacity to identify long-range dependencies within each sentence. With greater computational resources, an alternative configuration where the Bi-LSTM layer preceded the convolution layer, would've been possible.

In summary, while both models displayed competitive performances, TF-IDF with SVM came out to be the preferable option, attributed to its superior performance, ease of implementation, deployment, quicker training and testing, and its straightforward methodology.

6 Conclusion & Limitations

Moving forward, advancements in relation extraction could potentially benefit from exploring varied preprocessing techniques and hyperparameter optimization. The constraint of computational resources in our current study prevented us from applying extensive experimentation, especially with the more resource-intensive deep learning Bi-LSTM model. Also, the performance disparities observed in classes 10 and 18 highlight the need for in-depth pattern analysis and targeted feature engineering. In both our models, we excluded the underrepresented class 7 from the dataset, however, advanced data augmentation techniques could facilitate more effective training for sparse categories with limited data availability.

Incorporating manual spell checking and expanding the consideration for domain-specific vocabulary could further improve the quality of the model. Additionally, employing context-aware embeddings like BERT or ELMo could potentially enhance the model's semantic understanding. Addressing the limitations of GloVe embeddings with out-of-vocabulary terms is also something worth considering.

References

Akbani, R., Kwek, S. and Japkowicz, N. (2004) ‘Applying Support Vector Machines to Imbalanced Datasets’, in, pp. 39–50. Available at: https://doi.org/10.1007/978-3-540-30115-8_7.

Chawla, N. V. et al. (2002) ‘SMOTE: Synthetic Minority Over-sampling Technique’, Journal of Artificial Intelligence Research, 16, pp. 321–357. Available at: https://doi.org/10.1613/jair.953.

Dadgar, S.M.H., Araghi, M.S. and Farahani, M.M. (2016) ‘A novel text mining approach based on TF-IDF and Support Vector Machine for news classification’, in 2016 IEEE International Conference on Engineering and Technology (ICETECH). IEEE, pp. 112–116. Available at: https://doi.org/10.1109/ICETECH.2016.7569223.

Das, B. and Chakraborty, S. (2018) ‘An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation’. Available at: https://doi.org/10.48550/arXiv.1806.06407.

Detroja, K., Bhensdadia, C.K. and Bhatt, B.S. (2023) ‘A survey on Relation Extraction’, Intelligent Systems with Applications, 19, p. 200244. Available at: https://doi.org/10.1016/j.iswa.2023.200244.

GuoDong, Z. et al. (2005) ‘Exploring various knowledge in relation extraction’, in Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL ’05. Morristown, NJ, USA: Association for Computational Linguistics, pp. 427–434. Available at: https://doi.org/10.3115/1219840.1219893.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., ... Szpakowicz, S. (2010) 'SemEval-2010 Task 8: Multi-way classification of semantic relations between nominals'. In: Proceedings of the 5th International Workshop on Semantic Evaluation, Association for Computational Linguistics, pp. 33-38.

Hettinger, L. et al. (2018) ‘ClaiRE at SemEval-2018 Task 7: Classification of Relations using Embeddings’, in Proceedings of The 12th International Workshop on Semantic Evaluation. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 836–841. Available at: https://doi.org/10.18653/v1/S18-1134.

Hu, Y. et al. (2021) ‘Misspelling Correction with Pre-trained Contextual Language Model’. Available at: https://doi.org/10.48550/arXiv.2101.03204.

Jason Brownlee (2020) Cost-Sensitive SVM for Imbalanced Classification, https://machinelearningmastery.com/cost-sensitive-svm-for-imbalanced-classification/.

Jin, Y., Wu, D. and Guo, W. (2020) ‘Attention-Based LSTM with Filter Mechanism for Entity Relation Classification’, Symmetry, 12(10), p. 1729. Available at: https://doi.org/10.3390/sym12101729.

Kulshretha, S. & Lodha, L., 2023. Performance Evaluation of Word Embedding Algorithms. *International Journal of Innovative Science and Research Technology,* 8, pp.1555-1561. Available at: https://doi.org/10.5281/zenodo.10443962.

Liu, B., Wang, X., Xu, R., & Tang, B. (2014). Protein remote homology detection by combining profile-based protein representation with local alignment kernel. Journal of Medical and Bioengineering, 3(1), 17-22. https://doi.org/10.12720/jomb.3.1.17-22

Luthfi Ramadhan (2021) TF-IDF Simplified A short introduction to TF-IDF vectorizer, https://towardsdatascience.com/tf-idf-simplified-aba19d5f5530.

Ojokoh, B., Igbe, T., Afolabi, B., & Daramola, O., 2023. A graph model with integrated pattern and query-based technique for extracting answers to questions in community question answering system. *Social Network Analysis and Mining*, 13(1), p.45. Available at: https://doi.org/10.1007/s13278-023-01046-3.

Patle, A. and Chouhan, D.S. (2013) ‘SVM kernel functions for classification’, in 2013 International Conference on Advances in Technology and Engineering (ICATE). IEEE, pp. 1–9. Available at: https://doi.org/10.1109/ICAdTE.2013.6524743.

Qin, H., Tian, Y. and Song, Y. (2021) ‘Relation Extraction with Word Graphs from N-grams’, in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 2860–2868. Available at: https://doi.org/10.18653/v1/2021.emnlp-main.228.

Rink, B. and Harabagiu, S. (2010) UTD: Classifying Semantic Relations by Combining Lexical and Semantic Resources. Association for Computational Linguistics. Available at: http://wordnet.princeton.edu/.

rmdort (2017) spellchecker, https://github.com/rmdort/spellchecker
Available at: https://github.com/rmdort/spellchecker

Sawicki, J., Ganzha, M. and Paprzycki, M. (2023) ‘The State of the Art of Natural Language Processing—A Systematic Automated Review of NLP Literature Using NLP Techniques’, Data Intelligence, 5(3), pp. 707–749. Available at: https://doi.org/10.1162/dint_a_00213.

Singhania, S., Razniewski, S. and Weikum, G. (2022) ‘Predicting Document Coverage for Relation Extraction’, Transactions of the Association for Computational Linguistics, 10, pp. 207–223. Available at: https://doi.org/10.1162/tacl_a_00456.

Rossiello, G., Gliozzo, A., Farrell, R., Fauceglia, N. & Glass, M., 2019. Learning Relational Representations by Analogy using Hierarchical Siamese Networks. In J. Burstein, C. Doran & T. Solorio, eds. *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).* Minneapolis, Minnesota: Association for

Computational Linguistics, pp. 3235–3245. Available at: https://aclanthology.org/N19-1327. DOI: 10.18653/v1/N19-1327.

Zeng, D., Liu, K., Lai, S., Zhou, G. & Zhao, J., 2014. Relation Classification via Convolutional Deep Neural Network. In J. Tsujii & J. Hajic, eds. *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, pp. 2335–2344. Available at: https://aclanthology.org/C14-1220.

Zhang, S., Zheng, D., Hu, X. & Yang, M., 2015. Bidirectional Long Short-Term Memory Networks for Relation Classification. In H. Zhao, ed. Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation. Shanghai, China, October. pp. 73–78. Available at: https://aclanthology.org/Y15-1009 .

Zhao, K. et al. (2021) 'Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction', Knowledge-Based Systems, 219, p. 106888. Available at: https://doi.org/10.1016/j.knosys.2021.106888.

Appendices

A.

Relation Names	Label
Cause-Effect(e1, e2)	0
Cause-Effect(e2, e1)	1
Component-Whole(e1, e2)	2
Component-Whole(e2, e1)	3
Content-Container(e1, e2)	4
Content-Container(e2, e1)	5
Entity-Destination(e1, e2)	6
Entity-Destination(e2, e1)	7
Entity-Origin(e1,e2)	8
Entity-Origin(e2, e1)	9
Instrument-Agency(e1,e2)	10
Instrument-Agency(e2,e1)	11
Member-Collection(e1,e2)	12
Member-Collection(e2,e1)	13
Message-Topic(e1, e2)	14
Message-Topic(e2, e1)	15
Product-Producer(e1, e2)	16
Product-Producer(e2, e1)	17
Other	18

Table 1: Relation-Label.

B.

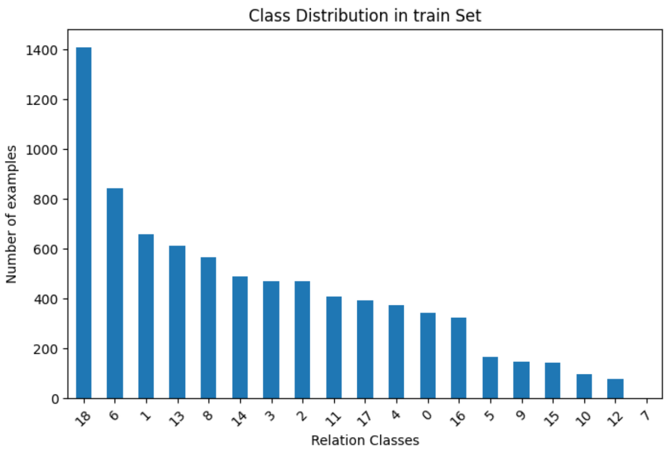

Figure 2: Class Distribution in train set

C.

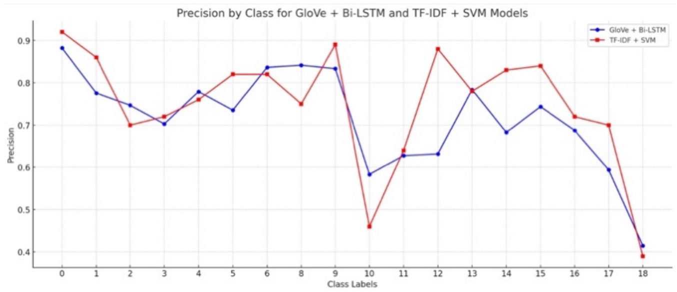

Figure 3: Precision by class for GloVe + Bi-LSTM and TF-IDF + SVM Modules

D.

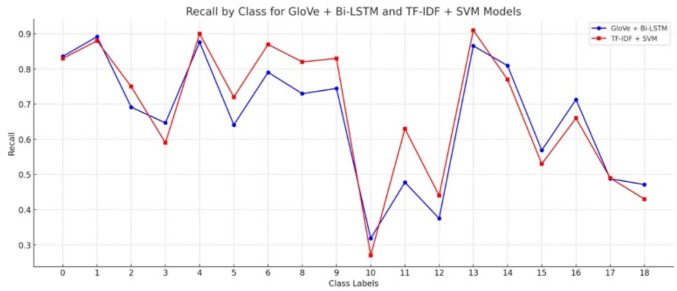

Figure 4: Recall by class for GloVe + Bi-LSTM and TF-IDF + SVM Models

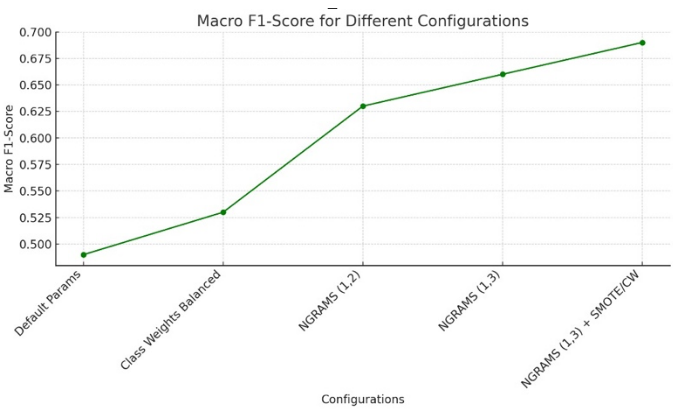

Figure 5: F1 score for different configurations