Amrita_CEN_NLP @ WOSP 3C Citation Context Classification Task

Premjith B, Soman K.P
Center for Computational Engineering and Networking (CEN)
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
b_premjith@cb.amrita.edu, kp_soman@amrita.edu

Abstract

Identification of the purpose and influence of citation is significant in assessing the impact of a publication. '3C' Citation Context Classification Task in Workshop on Mining Scientific Publication is a shared task to address the aforementioned problems. This working note describes the submissions of Amrita_CEN_NLP team to the shared task. We used various supervised learning algorithms for the for classification of sentences encoded into a vector of dimension 300 generated using Word2vec model.

1 Introduction

The number of publications in the scientific domain increased exponentially recently, which allows researchers to look for various literature to extend their research. One method of finding the more relevant literature is to check the number of citations. A publication with more citation generally has more influence in the research community. Such publications typically give significant insight into specific problems. To test whether a paper is relevant for a particular domain, one should analyse the context in which it is written. It is equally important to identify the context of the citations also. This information, as well as the citation count, give a better understanding of a publication in a particular domain. In (Pride and Knoth, 2017), David Pride and Petr Knoth proposed an automatic method for identifying the citations with influence. In addition to it, identification of the purpose of a citation is also an essential task.

This paper describes the submission of Amrita_CEN_NLP team in '3C' Citation Context Classification Task a part of Workshop on Mining Scientific Publications (WOSP) 2020 (N. Kunnath et al., 2020). This shared task consisted of two subtasks. The goal of Subtask-A was to identify the purpose of the citations. The Subtask-B intended to classify the classification based on their importance into either influential or incidental. We used machine learning-based models for identifying the purpose and influence of the citations according to the context. The Word2Vec (Mikolov et al., 2013b), (Mikolov et al., 2013a) algorithm was used to convert the words into vectors by capturing the contexts of words in the given corpus. We employed various classification algorithms with varying dimensions of word vectors for the aforementioned tasks. The Random Forest classifier (Liaw et al., 2002), (Premjith et al., 2019a), (Premjith et al., 2019b) with a word vector of size 300 achieved the best performance with 5-fold cross-validation.

The organization of the paper as follows: Section 2 gives an brief description on the related research, which will be followed by a description of the dataset in the Section 3. The next section discusses the steps involved in designing the machine learning model and the paper concludes with the Conclusion section.

2 Related Works

The number of research works reported for the classification of scientific literature according to the context in which it is written are limited despite it's significance. S. Teufel et al. (Teufel et al., 2006) proposed an annotation scheme along with a classification model for the categorization of the citations. They considered 12 classes for annotation. The machine learning model was trained over 2829 citation instances collected from 116 articles. They used IBK algorithm for classification with hand-engineered features like cue phrases. D. Jurgens (Jurgens et al., 2018) used feature such as pattern-based features, topic-based features, and prototypical argument features to classify the documents into 6 classes. The authors used Random

Forest algorithm for classification. A. Cohan et al. (Cohan et al., 2019) used Glove and ELMO word embedding features and Bidirectional LSTM with attention models for classifying the citations.

3 Dataset description

The training and test datasets used for Subtask-A and Subtask-B were the same. The training data and test data contained 3000 and 1000 sentences, respectively. The Subtask-A was a multiclass problem with six classes in it. The distribution of the data for this task was highly uneven. In the dataset, 54.93% of the data belong to the BACKGROUND category, whereas the share of the FUTURE category was mere 2.07%. The Subtask-B was a binary classification problem, and the data set for this task contained evenly distributed class labels.

4 Dataset description

The training and test datasets used for Subtask-A and Subtask-B were the same. The training data and test data contained 3000 and 1000 sentences, respectively. The Subtask-A was a multi-class problem with six classes in it. The distribution of the data for this task was highly uneven. In the dataset, 54.93% of the data belong to the BACKGROUND category, whereas the share of the FUTURE category was mere 2.07%. The Subtask-B was a binary classification problem, and the data set for this task contained evenly distributed class labels.

5 System description

Amrita_CEN_NLP team participated in both the subtasks. We used machine learning algorithms for both tasks. The implementation pipeline is as follows,

1. Preprocessing
2. Feature representation
3. Classification and Result analysis

5.1 Preprocessing

The first step in preprocessing was to remove the unwanted characters. Therefore, we removed all the characters other than alphabets and digits from the text. It is followed by converting all the characters into lower case. From this text, all the stop words were removed.

5.2 Feature representation

This work utilized the Word2Vec algorithm for representing the words as vectors. Initially, the pre-trained model, namely ”word2vec-google-news-300” was used for generating the word vectors. But the pre-trained model didn’t yield any good results with classification algorithms. Therefore, we decided to construct the vector representation out of the training and testing data. The input data for Word2Vec was constructed by combining both training and test data. We experimented with different embedding dimensions with Continuous Bag-of-Words training approach. The context window size was set to 5. The minimum number of occurrence of each word to be considered for word vector generation was assigned to 1 to make sure that all the words in the corpus will find a representation. The embedding dimensions considered for the experiment were 50, 100 and 300.

The sentence vector was constructed by taking the linear combination of the word vectors, where the coefficients were assigned to one.

5.3 Method

We used machine learning algorithms such as Decision Tree, Random Forest, K-Nearest Neighbor (KNN), AdaBoost, and Logistic Regression for classification, and analyzed their performance. The ultimate goal of the classification algorithms is to find the optimal parameters, which depends on the proper tuning of the hyper-parameters. To find the optimal set of hyper-parameters for a classifier for each subtask, we utilized GridSearchCV() defined in the scikit-learn Python package (Pedregosa et al., 2011). This function finds the best combination of hyper-parameters by implementing 5-fold cross-validation. This process was repeated for each classifier with different word embedding dimension. Table 1 shows the hyper-parameters used for tuning all the classifiers and the optimal parameters obtained after the hyper-parameter tuning. The first value in the third column represents the optimal hyper-parameter values used for Subtask-A, and the second value is used for Subtask-B. The best estimator was used for training the data and fixed the performance by again cross-validating with 5-folds. The imbalance in the dataset used for Subtask-A may cause the classifier to predict the class labels for test data biased towards the BACKGROUND class because of its percentage of share in the dataset.

Classifier	**Parameter**	**Parameter value**	**Optimal value**
Decision Tree	Splitting criterion	gini, entropy	entropy, gini
	Splitter	best, random	random, random
Random Forest	# Estimators	50,100,150	100, 50
	Splitting criterion	gini, entropy	gini, gini
	Maximum features	auto, sqrt, log2	sqrt, log2
KNN	# Neigbours	3,5,7	7, 5
	Weights	uniform, distance	uniform, uniform
	Algorithm	auto, ball_tree, kd_tree, brute	auto, auto
Adaboost	Learning rate	0.01, 0.1, 1, 10, 100	0.01, 0.1
	Algorithm	SAMME, SAMME.R	SAMME, SAMME
Logistic Regression	Penalty	l1, l2, elasticnet, none	l1 , l1
	C	0.01, 0.1, 1, 10, 100	0.01 , 0.01
	Solver	newton-cg, lbfgs, liblinear, sag, saga	liblinear , saga
	Multi class	auto, ovr, multinomial	auto , auto

Table 1: Set of hyperparameters used for training the classifiers

Classifier	**Embedding dimension**		
	50	**100**	**300**
Decision Tree	36.76	33.49	35.63
Random Forest	47.73	48.56	54.93
KNN	48.76	48.13	50.00
Adaboost	54.93	54.93	54.93
Logistic Regression	54.93	54.93	54.93

Table 2: Cross-validated results for identifying the purpose of citations

Classifier	**Embedding dimension**		
	50	**100**	**300**
Decision Tree	49.87	50.03	49.63
Random Forest	48.07	48.77	54.83
KNN	50.23	49.27	52.26
Adaboost	52.26	52.27	50.33
Logistic Regression	52.37	52.40	53.03

Table 3: Cross-validated results for identifying citation influence

The performances of the cross-validated models were evaluated using the accuracy score. The evaluation scores of identifying the purpose and influence of citations are given in Table 2 and Table 3

5.4 Result Analysis

From the Tables 2 and 3, it is clear that the feature vector with dimension 300 achieved the highest accuracy in both the tasks. For the Subtask-A, AdaBoost, Random Forest and Logistic Regression obtained the maximum classification accuracy of 54.93%. For the Subtask-B, Random Forest attained the highest accuracy of 54.83%. Therefore, we decided to submit the Random Forest model for both the shared tasks.

Performance of the models with test data was evaluated using F1-score (macro). Tables 4 and 5 show the public and private macro F1-scores. For identifying the purpose of citation, the Decision Tree algorithm achieved the highest public F1-score of 0.2071, whereas Logistic Regression obtained the private highest F1-score of 0.1953. Random Forest reported the highest public as well as private F1-scores for identifying he citation influence task.

6 Conclusion

This working note reports the submission of the team Amrita_CEN_NLP for both Subtask-A and

Classifier	Public	Private
Decision Tree	0.2071	0.1673
Random Forest	0.1780	0.1398
KNN	0.1662	0.1356
Adaboost	0.1205	0.1149
Logistic Regression	0.1731	0.1953

Table 4: Public and private F1-score (macro) for identifying the purpose of citations with best classifier

Table 5: Public and private F1-score (macro) for identifying citation influence

Classifier	Public	Private
Decision Tree	0.4757	0.4760
Random Forest	0.4894	0.5153
KNN	0.4639	0.4377
Adaboost	0.3046	0.3224
Logistic Regression	0.3125	0.3258

Subtask-B. We experimented with different classifiers and different word embedding dimensions for identifying the best model for the classification. The cross-validated results showed that Random Forest classifier with 300 dimension Word2Vec features achieved the highest accuracy for both shared tasks.

References

Arman Cohan, Waleed Ammar, Madeleine Van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. *arXiv preprint arXiv:1904.01608*.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association for Computational Linguistics*, 6:391–406.

Andy Liaw, Matthew Wiener, et al. 2002. Classification and regression by randomforest. *R news*, 2(3):18–22.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Suchetha N. Kunnath, David Pride, Bikash Gyawali, and Petr Knoth. 2020. Overview of the 2020 wosp 3c citation context classificationtask. In *The 8th International Workshop on Mining Scientific Publications*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

B Premjith, KP Soman, M Anand Kumar, and D Jyothi Ratnam. 2019a. Embedding linguistic features in word embedding for preposition sense disambiguation in englishmalayalam machine translation context. In *Recent Advances in Computational Intelligence*, pages 341–370. Springer.

Bhavukam Premjith, Kutti Padannayl Soman, and Prabaharan Poornachandran. 2019b. Amrita_cen@ fact: Factuality identification in spanish text. In *IberLEF@ SEPLN*, pages 111–118.

David Pride and Petr Knoth. 2017. Incidental or influential?–a decade of using text-mining for citation function classification.

Simone Teufel, Advaith Siddharthan, and Dan Tidhar. 2006. Automatic classification of citation function. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 103–110.