International Journal of Engineering & Technology, 8 (1.9) (2019) 237-242

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Utilizing Word Vector Representation for Classifying Argument Components in Persuasive Essays

Derwin Suhartono[1,2*], Afif Akbar Iskandar[2], M. Ivan Fanany[2], Ruli Manurung[2]

1Computer Science Department, Bina Nusantara University, Jakarta, Indonesia
2Machine Learning & Computer Vision Laboratory, Faculty of Computer Science, Universitas Indonesia, Depok, Indonesia
**Corresponding author E-mail: dsuhartono@binus.edu*

Abstract

Aside from the proper usage of grammar, diction and punctuation, a good essay must have cohesion and coherence. In persuasive essay, argumentative discourse is important as the parameter to see the cohesion and coherence among the arguments. An argument is characterized by one's stance (claim) which is strengthened with facts (premises) to complete the validity of the stance. Ideally, claims must be followed by premises either they support or attack the claims. In this paper, we try to identify 4 kinds of argument components (major claim, claim, premise, and non-argumentative) using some predefined features and measure the performance of word vector representation utilization in identifying argument components. We also present the results of our initial experiment by using deep learning to classify the argument components.

Keywords: *argument component, word vector representation, deep learning*

1. Introduction

Writing essay is not an easy task for most students either in a school or a university. It is because there are many rules which the authors should follow to form a well-written essay. Narrative, descriptive, expository, and persuasive are some categories which are defined to differ essays. Persuasive essay makes the writer must thoroughly research the topic, look at it from various perspectives, collect facts that should serve as solid evidence, and choose a side on the issue (LB Brief Handbook, 2013). Thus, writing a persuasive essay is quite challenging. If there are many components in an essay which describe an argumentation, the essay is supposed to be a persuasive essay.

The way how the authors arrange the argumentation structure become an important issue to produce a persuasive essay. An argument consists of some components and exhibits a structure which is based on the argumentative relation between components (Peldszus & Stede, 2013). An argument component contains a claim which is either supported or attacked by at least one premise (Stab & Gurevych, 2014b). The claim itself is central to an argument. Without any data in the form of a premise, the claim is a controversial statement.

There are many researches to classify argument components. Legal texts, persuasive essays, public policies, and essay scoring are some fields which are involved. They are described further in the next section of this paper. On the other hand, deep learning approach has successfully been applied to some text processing tasks such as sentence classification (Kim, 2014) and many others. Furthermore, word embedding as the result of deep learning architecture becomes the promising feature to be utilized. To the best of our knowledge, there is only one effort to utilize word embedding as the feature to classify argument component which was done by Stab and Gurevych (2016), yet there is no attempt on using deep learning approach to classify the argument components.

In this work, we did 3 (three) experiments. First, we implement most of the features by Stab & Gurevych (2014b) in classifying argument components. We investigate the result in the first experiment to be compared with the second experiment. In our second experiment, we try to utilize word embedding as the features in classifying argument components. We use Glove pre-trained word vectors (Pennington, Socher & Manning, 2014) for the word embedding feature. We did some comprehensive analysis regarding those two experiments. As the third experiment, we do an initial experiment in measuring the performance of argument components classification by using one of the deep learning framework named LSTM (Long Short Term Memory).

2. Argumentation Mining

There has been intensive research recently which try to identify and classify arguments into some categories. Some of them identify the relationship as well. Experiments to detect argument component in legal texts were successfully conducted (Moens, Boiy, Palau & Reed, 2007). In addition to the features they use, they utilize some features adopted from keyword list in Knott & Dale (1993). For the similar task, rule-based and probabilistic sequence model were combined automatically to detect the high-level organizational element in an argumentative discourse (Madnani, Heilman, Tetreault & Chodorow, 2012).

Slightly different from the research described in the previous paragraph, arguments for supporting public policy formulation were extracted (Florou, Konstantopoulos, Kukurikos & Karampiperis, 2013). The research result can help policymakers in knowing how is the reaction after the policy has been announced to the society. Tense and mood become the key features to indicate arguments in the experiments.

A rule-based algorithm to label each sentence in an essay with at most one label from our target argument ontology was also utilized as the approach (Ong, Litman & Brusilovsky, 2014). The approach used 8 rules to identify arguments in every sentence. Not only for identifying arguments, but they also used another 5 rules to do automated essay scoring as well. All rules were defined based on the intuition of the authors. In a similar yet different approach, argumentation scheme was implemented to score essays (Song, Heilman, Klebanov & Deane, 2014). The argumentation scheme was adopted from Walton's theory (Walton, 1996).

Due to a limited number of corpus in argumentation which had already been annotated, 90 persuasive essays were collected and annotated into 4 categories of argumentation labeling (Stab & Gurevych, 2014a). The annotation scheme is described in Figure 1.

In Figure 1, there are 3 (three) types of argument component; they are major claim, claim, and premise. They have the relationship between one to another. Premise supports or attacks claim and major claim. The claim is also utilized to confirm the existence of a major claim. Major claim and the claim do not have meaning if they are not supported by the premise. Statements which do not have any argument are labeled as non-argumentative. To ensure a good understanding of those 4 (four) argument components, we provide some examples of them. The following examples are taken from corpus (Stab & Gurevych, 2014a):

- Major Claim (MC):
  Newspapers have lost their competitive advantage to sustain their prolonged existence.
- Claim (C)
  The print media has failed to keep its important role in the provision of information.
- Premise (P)
  The Internet has been more and more popular for recent years, providing people with a huge source of information.
- None (N)
  As a result of this, print media such as newspapers have experienced a dramatic decline in the number of readers.

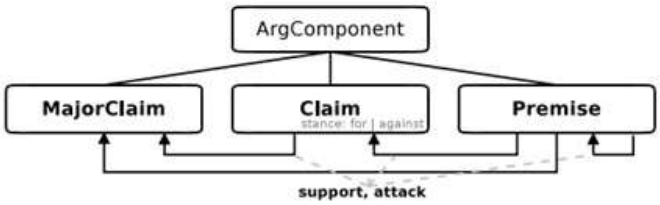

Fig. 1: Argument Component Annotation Scheme (Stab & Gurevych, 2014a)

After having done with the argument annotation, the corpus is utilized to automatically identify 4 (four) classes of argument component (Stab & Gurevych, 2014b). Features used to classify the argument components consist of structural, lexical, syntactic, indicator and contextual features.

As further exploration, we utilize 90 persuasive essays described before as well as some of the predefined features. Different with the previous research, we utilize various dimensions of word vector representation which is the result of deep learning to classify argument components in persuasive essays.

3. Deep Learning and word vector representation

By using deep learning, we expect to transform raw input data into higher level representations rather than lower level representations which are commonly obtained from conventional machine learning techniques. Deep learning comes from the initial thought related with human brains. Inspired by the architectural depth of brain, researchers try to train deep multi-layer networks (Bengio & LeCun, 2007). Some functions cannot be efficiently represented by using shallow architecture, but they can be represented efficiently by using deep learning.

The success story of deep learning started in image and speech processing fields. One of many deep learning models which are commonly used is Convolutional Neural Network (CNN). CNN was proposed to be utilized for face recognition (Lawrence, Giles, Tsoi & Back, 1997). Convolution happens when a large size matrix is multiplied with a small size matrix so that all values in the large one will be changed. By using CNN, the low-level feature can be transformed into the high-level feature. Features which characterize an image can be found in localized. After succeeding in face recognition field, this model was enhanced in handwriting recognition task (LeCun, Bottou, Bengio & Haffner, 2013).

There was an experiment using CNN in sentence classification (Kim, 2014). Even though this research is about text processing, there is no description involving linguistic knowledge. The experiment used 7 (seven) corpora collected as direct inputs to the CNN model. Word2Vec was utilized as the features to be trained in the model (Mikolov, Stuskever, Chen, Corrado, & Dean, 2013). Word2Vec is a word vector representation which is developed as universal feature extractor. It contains words along with their vectors which can be used in many tasks related to the text. Somehow, the vectors seem to represent the meaning of the words. The research result shows that pre-trained vector can yield a good performance. It concludes that word vector representation is an important ingredient to do an experiment in text processing especially when the task is closely related to deep learning.

Before Word2Vec appeared as a beneficial feature, a language modeling which was the origin of deep learning research in Natural Language Processing was built (Bengio, Ducharme, Vincent & Jauvin, 2003). Neural probabilistic language model was constructed to handle

word sequence that rarely or never been seen before. Aside from Word2Vec, another word vector representation as the result of deep learning named Glove was delivered (Pennington, Socher & Manning, 2014). An example of Glove vector from the word "and" is described as follows:

and 0.26818 0.14346 -0.27877 0.016257 0.11384 0.69923 -0.51332 -0.47368 -0.33075 -0.13834 0.2702 0.30938 -0.45012 -0.4127 -0.09932 0.038085 0.029749 0.10076 -0.25058 -0.51818 0.34558 0.44922 0.48791 -0.080866 -0.10121 -1.3777 -0.10866 -0.23201 0.012839 -0.46508 3.8463 0.31362 0.13643 -0.52244 0.3302 0.33707 -0.35601 0.32431 0.12041 0.3512 -0.069043 0.36885 0.25168 -0.24517 0.25381 0.1367 -0.31178 -0.6321 -0.25028 -0.38097

The words amount which were mapped to the word vector representation in Glove is 400,000.

4. Predefined features versus pre-trained word vectors

After having done in combining some features to detect argument components (Suhartono, 2015), this research tries to utilize Glove word vector representation as the feature to replace n-gram as the language model. By using Glove, the number of features is automatically reduced, and data sparsity is diminished. When we used n-gram, the number of word combination was very huge, yet they were rarely occurred in the corpus. In contrast, by using Glove, number of features is relatively small. It depends on which dimension is chosen as the pre-trained word vector. Glove provides some options regarding the word vector dimension.

For the experiment, we use the corpus of persuasive essays (Stab & Gurevych, 2014a). There are 90 essays which has been annotated so that we can parse to get the argument components list. The argument components are classified to major claim, claim, premise, and non-argumentative. To observe further whether word vector representation is good or not in classifying argument components, we prepare 2 (two) experiments. The first experiment is to implement most of the predefined features (Stab & Gurevych, 2014b) combined with discourse markers (Knott & Dale, 1993). The second experiment is to replace n-gram feature in first experiment with Glove word vector representation.

4.1. First Experiment

Features which are extracted from the corpus consists of 4 (four) general categories; they are structural, lexical, syntactic, and indicator. Contextual feature is not yet implemented like Stab & Gurevych (2014b) did. Previously, they use 55 discourse markers from the Penn Discourse Treebank 2.0 Annotation Manual (Prasad *et al.*, 2007) yet we use 286 discourse markers (Knott & Dale, 1993).

- Structural features
  - Number of tokens in covering sentence
  - Boolean feature determining whether argument component covers all tokens in the covering sentence or not
  - Number of tokens in the argument component
  - Number of tokens preceding and following the argument component
  - Number of punctuation in the covering sentence
  - Ratio of tokens between argument component and covering sentence
  - Boolean feature determining whether the sentence is ended by question mark or not
- Lexical features
  - Boolean feature occurrence of unigram
  - Boolean feature occurrence of bigram
  - Boolean feature occurrence of trigram
  - Boolean feature occurrence of modal
- Syntactic features
  - Number of sub-clauses included in the covering sentence
  - Depth of parse tree
  - Boolean feature of the production rule occurrence
- Indicator

  Some examples of the discourse markers (Knott & Dale, 1993) are:
  actually, by comparison, either, in this way, in conclusion
  The discourse markers are used as the features. They consist of 2 features as follows:
  - Number of discourse markers occurrence in the sentence
  - Boolean feature of the discourse markers

4.2. Second Experiment

In the second experiment, almost of the features presented in the first experiment are implemented. However, n-gram feature is removed and it is replaced by Glove word vector representation. The vector is placed alongside with the other features. This experiment is made to measure how well the pre-trained word vector works as the features to classify argument components.

5. Results and Discussion

All essays in the corpus are pre-processed into many argument components as defined in the annotation scheme (Stab & Gurevych, 2014a). By using all features which have been defined before, each argument component will contain some feature values.

Weka data mining software is used to quantify the performance of the features. Testing category uses 10-fold cross validation. Support Vector Machine (SVM) is used as the classifier. Afterwards, we have 3 different testing scenarios for utilizing word vector representation. Each scenario differs in the pre-trained word vectors dimensionality; they are 50, 100 and 200. Every argument component contains several words which means we will have some vectors in each argument component. We calculate the average value of those vectors, so that one argument component will have only one vector. For the experiment of utilizing word vector representation, all predefined fea-

tures are still used except N-gram language model. There are no significant differences if n-gram is replaced by Glove pre-trained word vectors as described in Table 1. Indeed, Glove with dimensional 100 has the same accuracy with N-gram. This indicated that Glove pre-trained word vectors do not have high significances in argument components classification. This fact is quite contradicting with the success of deep learning in other text processing tasks, e.g. sentence classification (Kim, 2014).

Table 1: Comparison Accuracy using N-gram Versus Pre-Trained Word Vector

Features	Attributes	Accuracy
N-gram	313	73.9311
Glove 50 dim.	63	73.8717
Glove 100 dim.	113	73.9311
Glove 200 dim.	213	73.5154

Weka provides some filters to select which features are representative to be used in classification tasks. One of them is attribute selection. In this experiment, we use attribute selection as the filter to all features. We use attribute selection to ensure how many features are categorized as good features to classify argument components. However, the number of selected features between n-gram language model and Glove pre-trained word vector has no significant differences. It indicates that n-gram is a good feature to classify argument components. Word vector representation has quite similar performance with n-gram. In term of accuracy, n-gram performs slightly better than Glove word vector representation as described in Table 2.

Table 2: Comparison Accuracy using N-gram Versus Pre-Trained Word Vector After Filtered by Attribute Selection

Features	Attributes	Accuracy
N-gram	10	75.2969
Glove 50 dim.	8	74.7031
Glove 100 dim.	9	74.7031
Glove 200 dim.	12	74.9406

To see more objectively the influence of n-gram compared with word vector representation in classifying argument components, we conduct other experiments by removing all features except n-gram and word vector representation.

N-gram attributes are built up from all possible word combination in corpus. We generate unigram, bigram, and trigram from the corpus and take top-100 of each so that the attributes become 300 in total. Surely the value in N-gram contains a big data sparsity. By utilizing less features and close accuracy values, Glove pre-trained word vector gives us the same accuracy in average compared with N-gram. The result is described in Table 3.

Table 3: Comparison Accuracy using N-gram Versus Pre-Trained Word Vector without Other Predefined Features

Features	Attributes	Accuracy
N-gram	300	52.9691
Glove 50 dim.	50	53.3254
Glove 100 dim.	100	53.2067
Glove 200 dim.	200	52.9691

Furthermore, we did a similar experiment which involved attribute selection as the feature filter. The results comparison is described in Table 4.

Table 4: Comparison Accuracy using N-gram Versus Pre-Trained Word Vector without Other Predefined Features after filtered by Attribute Selection

Features	Attributes	Accuracy
N-gram	9	54.2162
Glove 50 dim.	23	53.4442
Glove 100 dim.	37	53.2067
Glove 200 dim.	57	52.9691

By involving attribute selection as the filter, the number of significant Glove attributes is larger than N-gram. In term of its accuracy, Glove pre-trained word vector gives worse result than N-gram. In this scenario, we can conclude that features in word vector representation have better significances than N-gram. It is because the number of features in pre-trained word vector after filtered is relatively larger than N-gram.

If the confusion matrix is observed, we find out that the main issue is to detect major claim (MC) properly. From the whole scenarios, none of the major claim is well detected. Most correct classification is the premise (P). On the other hand, the accuracy to identify claim (C) is still very low. Therefore, we still need to look for definitive features in detecting major claim (MC) and claim (C).

6. Initial Experiment Using Deep Learning

We did an initial experiment for classifying argument components using deep learning as the third experiment. Again, we utilized Glove word vector representation as the data that is inputted to the deep learning architecture. Keras (http://keras.io/) is used as the neural network library to implement our experiment. This library is capable of running on top of TensorFlow and Theano, yet we used Theano.

There are many architectures in deep learning. In this experiment, we used two (2) layer LSTM (Long Short Term Memory) as one of variants in RNNs (Recurrent Neural Network). 50-dimensional word vectors from Glove was used as the data in the input layer. Due to four (4) categories that we have, we use categorical crossentropy for compiling the model. We utilized two (2) kind of activation functions; they are tanh and sigmoid with additional dense layer. Table 5 informs us the details of the experiment.

Table 5: Accuracy of Argument Component Classification using Categorical Crossentropy

LSTM Configuration	Activation Functions	
	Sigmoid	**Tanh**
using Dropout	Loss value: 2.6971 Accuracy: 54.80%	Val loss: 4.0191 Accuracy: 62.60%
without Dropout	Loss value: 3.95 Accuracy: 55.58%	Loss value: 13.1043 Accuracy: 12.47%
without Dropout, and initialize LSTM with Uniform	Loss value: 3.9253 Accuracy: 54.54%	Loss value: 12.987 Accuracy: 12.99%
without Dropout, and initialize dense layer with Uniform	Loss value: 4.6681 Accuracy: 57.40%	Loss value: 10.9687 Accuracy: 12.47%
without Droput, initialize LSTM with Uniform, and initialize dense layer with Uniform	Loss value: 4.882 Accuracy: 56.36%	Loss value: 10.9687 Accuracy: 19.48%

As the additional scenario number 4, we changed the dense layer to be glorotuniform which got 1.1871 as the loss value and 57.66% as the accuracy. Compared with another result, we conclude this is the best setting so far. The progress of each iteration shows a good learning process from the model. The strange condition appeared in scenario number 1 which uses tanh as the activation function. By observing the value in each iteration, the algorithm is considered not to learn anything, yet the accuracy is the highest among all (62.60%).

We find out that the experiment did not carry out a good result. For deep learning experiment, we guess that 90 essays are too small as the dataset. A larger size of data is recommended. Fortunately, Stab and Gurevych (2016) has published new corpus which contains 402 essays. Later, we will use this corpus to investigate the result of argument component classification by using deep learning.

7. Conclusion

Based on the experiment conducted on this research, we conclude that:

1. Glove pre-trained word vectors do not have high significances in improving the performance of argument components classification
2. Close gap between accuracy of n-gram and Glove pre-trained word vectors indicates that n-gram is good enough as the features to classify argument components rather than Glove pre-trained word vectors
3. Less number of features in pre-trained word vectors compared with n-gram shows us that using pre-trained word vectors in argument components classification will consume less memory usage
4. Glove word vector representation is not yet able to utilized for classifying major claim (MC) and claim (C) properly
5. Experiment in deep learning needs huge number of dataset, yet we only have 90 persuasive essays.

Some future works that seems visible to be done are:

1. Using Word2Vec as the universal feature extractor to replace Glove. Even further, we can compare the results between Word2Vec and Glove
2. Do further comprehensive and other experiments to use deep learning in classifying argument component
3. Utilize the new argument annotated essays which contains 402 persuasive essays (Stab & Gurevych, 2016)

References

[1] Bengio, Y., Ducharme, R., Vincent, P. and Jauvin, C. (2003). A Neural Probabilistic Language Model. Journal of Machine Learning Research, 3, pages 1137-1155.

[2] Bengio, Y. and LeCun, Y. (2007). Scaling learning algorithms towards AI. In Bottou, L., Chapelle, O., DeCoste, D., Weston, J. (Eds.), Large Scale Kernel Machines. MIT Press.

[3] Florou, E., Konstantopoulos, S., Kukurikos, A. and Karampiperis, P. (2013). Argument Extraction for Supporting Public Policy Formulation. Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, pages 49-54, Sofia, Bulgaria, August 8.

[4] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746-1751, October 25-29, Doha, Qatar.

[5] Knott, A. and Dale, R. (1993). Using Linguistic Phenomena to Motivate a Set of Rhetorical Relations. Technical Report HCRC/RP-39, Edinburgh, Scotland.

[6] Lawrence, S., Giles, C. L., Tsoi, A. C. and Back, A. D. (1997). Face Recognition: A Convolutional Neural-Network Approach. IEEE Transactions on Neural Networks, Vol. 8, No. 1, January.
[7] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (2013). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278-2324.
[8] Madnani, N., Heilman, M., Tetreault, J. and Chodorow, M. (2012). Identifying High-Level Organizational Elements in Argumentative Discourse. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 20-28, Montreal, Canada, June 3-8.
[9] Mikolov, T., Stuskever, I., Chen, K., Corrado, G. S. and Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. Proceedings of Neural Information Processing System (NIPS).
[10] Moens, M.F., Boiy, E., Palau, R. M. and Reed, C. (2007). Automatic Detection of Arguments in Legal Texts. The 11th International Conference on Artificial Intelligence and Law, June 4-8, Stanford Law School, Stanford, California.
[11] Ong, N., Litman, D. and Brusilovsky, A. (2014). Ontology-Based Argument Mining and Automatic Essay Scoring. Proceedings of the First Workshop on Argumentation Mining, pages 24-28, Baltimore, Maryland USA, June 26.
[12] Peldszus, A. and Stede, M. (2013). From Argument Diagrams to Argumentation Mining in Texts: A Survey. International Journal of Cognitive Informatics and Natural Intelligence (IJCINI), 7(1): 1-31.
[13] Pennington, J., Socher, R. and Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. Empirical Methods in Natural Language Processing (EMNLP), Volume 14, pages 1532-1543.
[14] Prasad, R., Miltsakaki, E., Dinesh, N., Lee, A., Joshi, A., Robaldo, L. and Webber, B. L. (2007). The Penn Discourse Treebank 2.0 annotation manual. Technical report, Institute for Research in Cognitive Science, University of Pennsylvania.
[15] Song, Y., Heilman, M., Klebanov, B. B. and Deane, P. (2014). Applying Argumentation Schemes for Essay Scoring. Proceedings of the First Workshop on Argumentation Mining, pages 69-78, Baltimore, Maryland USA, June 26.
[16] Stab, C. and Gurevych, I. (2014a). Annotating Argument Components and Relations in Persuasive Essays. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 1501-1510, Dublin, Ireland, August 23-29.
[17] Stab, C. and Gurevych, I. (2014b). Identifying Argumentative Discourse Structures in Persuasive Essays. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 46-56, Doha, Qatar.
[18] Stab, C. and Gurevych, I. (2016). Parsing Argumentation Structures in Persuasive Essays. arXiv preprint, arXiv: 1604.07370.
[19] Suhartono, D. (2015). Klasifikasi Komponen Argumen Secara Otomatis pada Dokumen Teks berbentuk Esai Argumentatif. arXiv preprint, arXiv:1512.00578.
[20] The Argumentative (Persuasive) Essay (2013). LB Brief Handbook, 5th edition, Chapter 12; OWL at Purdue website. Austin Peay State University.
[21] Walton, D.N. (1996). Argumentation Schemes for Presumptive Reasoning. Mahwah, NJ: Lawrence Erlbaum.