

## 基於有向圖與爭論導向摘要的網路辯論之爭論元素辨識

### Identifying Argument Components in Online Debates through Directed Graph and Argument-oriented Summarization

魏奇安 Chi-An Wei

國立成功大學資訊工程研究所

Institute of Computer Science and Information Engineering

National Cheng Kung University

[kevinwei3@gmail.com](mailto:kevinwei3@gmail.com)

高宏宇 Hung-Yu Kao

國立成功大學資訊工程研究所

Institute of Computer Science and Information Engineering

National Cheng Kung University

[hykao@mail.ncku.edu.tw](mailto:hykao@mail.ncku.edu.tw)

#### 摘要

辨識觀點在「辯論語意挖掘」的領域中扮演著重要的角色。在任意文章中辨識出論點，不僅可以用在被立場分類做為特徵來源增加辨識的準確性，同時也可以做為一篇文章表明其支持或反對一個目標的理由。在這項研究中，已經有許多方法嘗試解決這一項任務，較廣泛提起的是文字分類方法和自動化摘要的技巧。然而，將辨識觀點轉化成文字分類的問題不僅著重在特徵的選擇和使用，同時也因為將文章中一個一個句子拆出視為獨立的句子，所以也失去了文章整體的關係；自動化摘要的技巧雖然將文章中的所有句子一起處理評估出其中一句適合做為文章代表的句子，然而現有的一些方法多使用詞袋模型以及通用的自動化摘要方法，不僅沒有表示的特徵過於稀疏，同時亦沒有考量及引入辯論文章的特性。

本研究主要在可以保留文章為一體的自動化摘要方法上進行深刻的觀察及調整，使其不僅可以以辯論強度為導向摘要文章，並建構一有向圖來表示這些文章。在這些增強和調整下，不僅讓原本無辯論導向的自動化摘要方法可以更好的排序出辯論的句子。最終，本研究提出的方法可以在辯論文章中識別論點句子的任務上提升 8% 的辨識準確率。

## Abstract

Identifying argument components has become an important area of research in argument mining. When argument components are identified, they can not only be used for stance classification but also can provide reasons for determining an article is supporting or opposing about a specific target.

Previous research mainly used text classification and summarization techniques to solve this task. However, by transforming the task to a classification problem, not only rely heavily on choosing and using bag-of-words features, but also lose the article entity information due to extract the sentences out of the article and treat as an individual training instance. In the other hand, although summarization techniques handle on entire article and try to figure out which sentence can best represent the core concept of the article, in identifying argument components still heavily relies on bag-of-words feature representation and lack of argument-oriented features to concern about argument components characteristics.

In our study, we dive down to the core of the summarization method, not only makes it based on argument strength to summarize articles and identify argument components, but also proposed a directed graph construction approach. Experiments show that our proposed method outperforms 8% better than those without argument-oriented methods.

關鍵詞：辯論語意挖掘、辨識論點、自動化摘要、有向圖

Keywords: Argument Mining, Argument Components, Summarization, Directed Graph.

### 1. Introduction

In “Argument Components Extraction (ACE),” researchers try to get which sentence can be treated as arguing points, as known as “Argument Components (AC).” According to [1], which summarized tons of state-of-art knowledge about AM, summarized that how can a sentence become an argument component. The argument components in argumentative articles are usually formed by five types of sentences: “Claim” are the sentences that represent the statement being argued, “Data” are the facts or evidence used to prove the claim, “Warrant” are the sentences that make a connection between data and claim, “Backing” and “Rebuttal” are the sentences that support and against the warrant, respectively.

ACE is what our study is targeting on, to find out which are the key sentences that make

people explain why they support or against something. Furthermore, the results can feedback to SC task, since we know the key sentences that lead an article to support or against.

We are using the dataset released by [2], which collected posts under the domain “Abortion,” “Gay Rights,” “Obama” and “Marijuana” from an online debate forum. They manually labeled argument components in each post under each domain.

To accomplish the goal of ACE, previous methods commonly use bag-of-words or feature-based approach to represent the sentence in a feature vector form. These methods will cause the problems that lead to sparse feature, and it needs to treat sentences individually rather than processing with other sentences in the post.

We based on existed summarization method, TextRank to be one of our baselines. In [3], they aim to extract argument components by applying TextRank, which first using TF-IDF to represent each sentence then create an undirected graph, then applies PageRank on the undirected graph to acquire ranked sentences; In the end, the top-ranked sentence will be treated as argument components. However, we can say that the summarization algorithm was proved to perform well in extracting keywords or key sentences from an article, but cannot confidently say they are argument components.

In [4], which their work motivated us to integrate argument-oriented information in the graph, shows that changing the edge construction method can improve TextRank performance. We proposed an argument-oriented TextRank, ArguRank to address previously issued problems: With integrating subjectivity score to change the calculation within TextRank, we can confidently say that the result will be argument-ranked. Moreover, we proposed a directed graph construction approach to retrieve the nodes relation and direction, which aim to gain more performance by concern about how the score will be propagated.

To summarize, we make the following contributions:

- A model to retrieve subjectivity score of words through manual compiled argument-oriented corpus.
- An argument-oriented TextRank to identify the argument component.
- A directed graph construction approach to pursuing better ranking performance.

## 2. Related Works

### 2.1 Lexicon Expansion

The lexicon expansion approach aims to enlarge an existing lexicon using possibility, so it can not only be used by itself but can also lead a classifier or a model to learn its core concepts.

Previously, a lexicon-based method uses the attributes provided in the lexicon to do a specific task. The attribute in the lexicon usually becomes a feature that whether the target article contains the word or not, if contains, then the feature will switch to a state, otherwise will have another state. But more chances words in the target article will not appear in the chosen lexicon, make the feature is not as efficient as it should be. Researchers like [5] proposed a concept that using an existing lexicon as seed, and choose a model to learn the concept of the lexicon and then the model can try to predict attributes of words that not contain in the lexicon to maximize the feature that we want to retrieve from the lexicon. The lexicon expansion is accomplished in the following step, which proposed in [5]: (1) Choose a lexicon as seed, (2) Transfer words from text to its representation in vector space, finally (3) Construct and train a classifier or model to predict unknown words to retrieve the information the seed lexicon can give. The main workflow is shown in our research; we will use such this approach to enlarge an existing subjectivity lexicon to acquire words subjectivity strength probability for further procedure.

### 2.2 Graph-based Summarization

Another field and category to summarize documents are the graph-based methods. One of this kind is LexRank, proposed by [6]. It adopts TF-IDF to represent each words' importance in the sentences, then uses a modified cosine similarity equation to construct the edges between sentences.

The other approaches are TextRank and TextRank-based variation. [3] modified TextRank (shorten as PsTK) to rank sentences to determine which sentences are highly possible to be an argument component. The work constructs the graph for PageRank to iterate with sentences as nodes and similarity between each sentence as edges. In the original TextRank [7], it uses the following Equation 1 to calculate the similarity between sentences.

$$Similarity(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \& w_k \in S_j\}|}{\log(|S_i|) + \log(|S_j|)} \quad (1)$$

After the edges are calculated, TextRank will use PageRank, proposed by [8] to iterate the graph and get the scores of each node, here we treat nodes as sentences in an article. In PsTK, they use a different way to construct the edges. However, the method cannot convince that summarization is a suitable method to solve the task, since none of the operation related to argument, stance or reasoning. In our work, we proposed an argument-oriented TextRank, ArguRank, which considered argument specific characteristics to identify argument component.

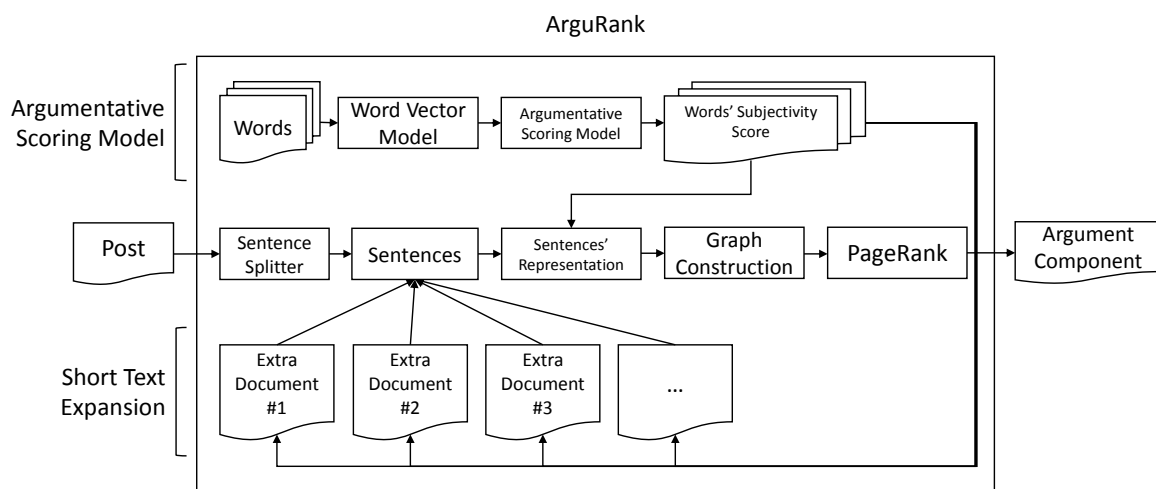


Figure 1. System framework and flowchart of our proposed method.

### 3. Method

We based on PsTK, the state-of-the-arts method on our dataset to develop our method, ArguRank. In the previous section, we address the issue of TextRank, motivated us to proposed argument-oriented TextRank, ArguRank, which aim to solve these issues and performs better in identifying argument components. Our proposed system will (1) read online debate posts, (2) preprocess the texts into sentences, (3) create sentence representation for calculating edge, (4) build a directed graph then (5) apply PageRank to scoring sentences. During this pipe which shows in Figure 1, we will apply our method to certain steps to make become argument-oriented TextRank, ArguRank.

### 3.1 Argumentative Scoring Model

To make TextRank argument-oriented, we develop a model to help enhance extract hidden argumentative information. The corpus we use to build the model is “Multi-Perspective Question Answering (MPQA) Subjectivity Lexicon,” was compiled by [9]. It contains 8222 words; each word was labeled as “strong subjectivity” or “weak subjectivity.” To get the score of words that are not contained in the lexicon, we use the word vector model and binary classifier to build an argumentative scoring model to predict the argumentative score of the word. To construct the argumentative scoring model, we (1) use a word vector model to get its vector representation, then (2) feed to a binary classifier to make it learn how to separate the vector into strong or weak subjectivity.

### 3.2 Argument-oriented Summarization

After training the classifier, we use it to get subjectivity score of each word in each sentence. First, we normalized the subjectivity score of words in each sentence by a softmax function. The sentence representation will then be calculated via Equation 2.

$$\vec{S} = \sum_i^K \vec{V}_{W_i} a_i \quad (2)$$

Where  $\vec{V}$  represents the word vector of  $W_i$  and  $\vec{S}$  denotes the sentence representation before constructing the graph for PageRank.

### 3.3 Graph Construction

After we acquire various sentence representations, the next step is to construct the graph that represents the article. We choose cosine similarity to retrieve sentence relationship as edges, which shows in Equation 3.

$$\hat{E}(\vec{S}_i, \vec{S}_j) = \frac{\vec{S}_i \cdot \vec{S}_j}{\|\vec{S}_i\| \cdot \|\vec{S}_j\|} = \frac{\sum_{k=1}^n \vec{S}_{i_k} \cdot \vec{S}_{j_k}}{\sqrt{\sum_{k=1}^n \vec{S}_{i_k}^2} \sqrt{\sum_{k=1}^n \vec{S}_{j_k}^2}} \quad (3)$$

where  $\vec{S}_i, \vec{S}_j$  are representations of two sentences.

After the edges been calculated, the graph  $G$  will be constructed.  $S_1, S_2, S_3$  denotes the sentences respectively, and  $E_{12}, E_{23}, E_{13}$  indicates the similarity between sentences.

Based on the undirected graph, we then further apply two conditions to make it becomes directional. First is the condition to determine what edges are going to be discarded due to low similarity, which determined by  $Threshold_{Dis}$  and shows in Equation 4.

$$E_{ij} = \begin{cases} \hat{E}(\vec{S}_i, \vec{S}_j), & \text{if } \hat{E}(\vec{S}_i, \vec{S}_j) \geq Threshold_{Dis} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Next, the construction of directional graph representation will be completed in two sub-steps. First, every sentence will have a direction that points to its next sentence, which shows in Equation 5.

$$E_{ij} = \begin{cases} E(\vec{S}_i, \vec{S}_j), & \text{if } j - i = 1 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Last, the sentences which have high similarity (determined by  $Threshold_{Sim}$ ) will be used to applied direction that makes two nodes points together, which similar to Equation 4 and shows in Equation 6.

$$E_{ij}E_{ji} = \begin{cases} \hat{E}(\vec{S}_i, \vec{S}_j), & \text{if } \hat{E}(\vec{S}_i, \vec{S}_j) \geq Threshold_{Sim} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where the difference is the threshold will be interpreted as how similar of the sentences will have a bi-directional connection. The directed graph can be visualized in Figure 2.

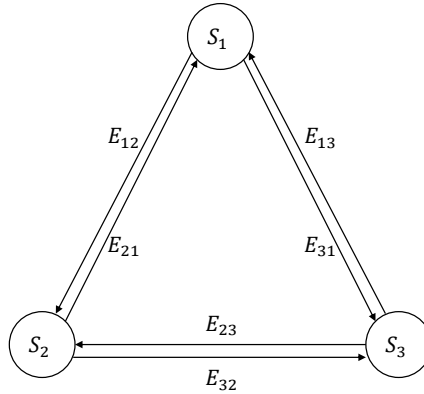


Figure 2. A directed graph that represents the article.

In the end, PageRank will applied on the directed graph to output the rank of these sentences. The top-1 ranked sentences will then be considered as argument components of the online debate article.

## 4. Experiments and Discussion

### 4.1 Evaluation Metrics

The methods we proposed required different evaluation metrics. For assessing the argumentative scoring model, first, we will use accuracy to evaluate how well the model can correctly predict whether the word in the lexicon that is strong or weak subjectivity.

Moreover, to prevent the illusion that the accuracy gives we adopt sensitivity (also called the true positive rate, or recall) and specificity (also called the false positive rate) to observe how the model performs on the answer in the strong and wrong subjectivity, respectively.

To evaluate ArguRank, we use accuracy again to assess how many articles in the dataset are correctly found the argument components. If the top-1 ranked sentence match one of the annotated argument components within an online debate article then the accuracy will raise.

### 4.2 Result Discussion

There are two main parts we are going to discuss in this section: (1) Building Argumentative Scoring Model and (2) Graph Construction.

#### 4.2.1 Building Argumentative Scoring Model

We select three word vector models: Word2vec, GloVe, and fastText for comparison. The result shows in Table 1, which using fastText and debate posts plus Wikipedia articles as training corpus plus the original lexicon words can get acceptable performance on predicting subjectivity score of words than other methods. The model than further being used in applying the score of each word in sentence representation.

Table 1. Final argumentative scoring model where boldfaced scores show that better than others.

|             | <b>Word2vec</b>     | <b>GloVe</b> | <b>fastText</b>     |
|-------------|---------------------|--------------|---------------------|
| Accuracy    | 0.776               | 0.767        | <b><u>0.777</u></b> |
| Sensitivity | <b><u>0.798</u></b> | 0.789        | 0.767               |
| Specificity | 0.729               | 0.719        | <b><u>0.798</u></b> |
| AUC         | 0.834               | 0.829        | <b><u>0.86</u></b>  |



## 4.2.2 Graph Construction

In constructing a directed graph, due to the sentence splitting operation, some articles in our dataset exist only one sentence, which cannot construct the graph since no enough nodes. After filtering these kinds of articles, the filtered datasize is shown in Table 2, also with the performance of various directed graph construction approaches. By only drop connections on the undirected graph, the accuracy will get slightly improvement, and the performance will get better after direction construction with different granularity.

Table 2. The results of different graph constructions using cosine similarity. The different number of size about valid dataset in use to construct graph is also shown in here.

|             | <b>ABO</b>          | <b>GAY</b>          | <b>OBA</b>          | <b>MAR</b>          | <b>avg</b>          |
|-------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| PsTK        | 0.469               | 0.528               | 0.582               | 0.641               | 0.555               |
| ArguRank    | <b><u>0.490</u></b> | <b><u>0.538</u></b> | <b><u>0.628</u></b> | <b><u>0.693</u></b> | <b><u>0.587</u></b> |
| Improvement | 0.021               | 0.01                | 0.046               | 0.052               | 0.032               |

## 4.2.3 Positions of Argument Components

We apply our method to the datasize that removes the posts which one of the argument components is in the first sentence and PsTK predicted. The reason to remove these posts is in PsTK or other methods based on undirected graph, it will face a problem if the ranked scores are the same, the algorithm will predict the first sentence of the post to be the argument components. To show how our directed graph construction approach can deal with such this problem, we experimented on the datasize after the removal. In Table 3, we run the directed graph construction methods on the datasize and filtered graph size; the results show that by representing sentence by specific weighting mechanism and the directed graph construction, our method can identify argument components better than PsTK.

Table 3. After removing the posts that one of its argument components is located at the first place of sentences.

|                     | <b>ABO</b>          | <b>GAY</b>          | <b>OBA</b>          | <b>MAR</b>          | <b>avg</b>          |
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| PsTK                | 0.430               | 0.475               | 0.512               | 0.568               | 0.496               |
| ArguRank            | 0.503               | 0.553               | 0.609               | 0.634               | 0.575               |
| TextRank + ArguRank | <b><u>0.515</u></b> | <b><u>0.548</u></b> | <b><u>0.606</u></b> | <b><u>0.637</u></b> | <b><u>0.577</u></b> |
| Improvement         | 0.085               | 0.073               | 0.094               | 0.069               | 0.081               |

## 5. Conclusions

We based on TextRank to develop an argument-oriented and directed ranking method called “ArguRank,” which makes TextRank argumentative and directed. Also, we show how we build our research environment to expand a lexicon for identifying argumentative words and construct an argument representation.

The experiments show the proof that using argument-oriented graph-based summarization method by applying the subjectivity lexicon to construct the sentence representation can get better result on extracting argument components. Moreover, the approach of directed graph construction significantly improves the performance of identifying argument components via graph-based summarization.

## References

- [1] da Rocha, G.F. *ArgMine: Argumentation Mining from Text*. 2016; Available from: <http://hdl.handle.net/10216/89719>.
- [2] Hasan, K.S. and V. Ng. *Why are you taking this stance? Identifying and classifying reasons in ideological debates*. in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014.
- [3] Petasis, G. and V. Karkaletsis. *Identifying argument components through TextRank*. in *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*. 2016.
- [4] Barrios, F., et al., *Variations of the similarity function of textrank for automated summarization*. arXiv preprint arXiv:1602.03606, 2016.
- [5] Bar-Haim, R., et al. *Improving claim stance classification with lexical knowledge expansion and context utilization*. in *Proceedings of the 4th Workshop on Argument Mining*. 2017.
- [6] Erkan, G.u., nes and D.R. Radev, *Lexrank: Graph-based lexical centrality as salience in text summarization*. *journal of artificial intelligence research*, 2004. **22**: p. 457-479.
- [7] Mihalcea, T., P.T. Textrank, and others. *Bringing order into texts*. in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2004.
- [8] Page, L., et al., *The PageRank citation ranking: Bringing order to the web*. 1999, Stanford InfoLab.
- [9] Wilson, T., J. Wiebe, and P. Hoffmann. *Recognizing contextual polarity in phrase-level sentiment analysis*. in *Proceedings of the conference on human language technology and empirical methods in natural language processing*. 2005.