Rhetorical Sentences Classification Based on Section Class and Title of Paper for Experimental Technical Papers

Afrida Helen*, Ayu Purwarianti & Dwi. H. Widyantoro

Department of Informatics, School of Electrical Engineering and Informatics (STEI)
Bandung Institute of Technology, Jalan Ganesha 10, Bandung, 40132, Indonesia
Email: helen@pens.ac.id

Abstract. Rhetorical sentence classification is an interesting approach for making extractive summaries but this technique still needs to be developed because the performance of automatic rhetorical sentence classification is still poor. Rhetorical sentences are sentences that contain rhetorical words or phrases. Rhetorical sentences not only appear in the contents of a paper but also in the title. In this study, features related to section class and title class that have been proposed in a previous research were further developed. Our method uses different techniques to reach automatic section class extraction for which we introduce new, format-based features. Furthermore, we propose automatic rhetoric phrase extraction from the title. The corpus we used was a collection of technical-experimental scientific papers. Our method uses the Support Vector Machine (SVM) algorithm and the Naïve Bayesian algorithm for classification. The four categories used were: *Problem*, *Method*, *Data*, and *Result*. It was hypothesized that these features would be able to improve classification accuracy compared to previous methods. The F-measure for these categories reached up to 14%.

Keywords: *classification; extraction; pattern-matching; preposition-based; rhetorical phrase; rhetorical sentence; section and title.*

1 Introduction

Automatic rhetorical sentence classification is a form of computational intelligence application that aims to extract sentences from a text document based on categories. By definition, a rhetorical sentence is the way the author conveys meaning to the reader. The technique of classification will be specific for each category. A scientific paper contains several types of rhetorical sentences with specific characteristics. The main characteristic of a rhetorical sentence is that it contains a word-phrase that belongs to the author. A summary can be generated automatically by collecting rhetorical sentences based on sentence categories [1,2]. This research concerns the automatic extraction of rhetorical sentences based on four categories. We add several new features to increase the accuracy of the automatic rhetorical sentence classification.

Received June 5th, 2015, 1st Revision October 16th, 2015, 2nd Revision December 11th, 2015, Accepted for publication December 13th, 2015.
 ISSN: 2337-5787, DOI: 10.5614/itbj.ict.res.appl.2015.9.3.5

Rhetorical sentences appear in the rhetorical divisions of a paper [3,4]. We have analyzed a number of technical-experimental papers (i.e. common scientific papers–APA and IEEE [5]) [6]. Their format is typical for technical-experimental papers. For scientific papers, a prototypical rhetorical division has been established, typically consisting of Introduction, Proposed Solution, Experimental Design, Result, Discussion, Conclusion, etc. This is especially the case for research texts from the exact sciences, where the rhetorical divisions tend to be marked very clearly with section headers [3]. This study used one abstract and five sections (Abstract, Introduction, Related Work, Method, Experimental Result, and Conclusion) to find places where rhetorical sentences appear.

The corpus of this research was a collection of experimental technical papers. We used this type of research papers because no studies have investigated experimental technical papers yet and therefore we introduce the rhetorical sentences that usually appear in this type of paper (i.e. *Problem* of the research, *Data* of the research object, the *Method* used, and the *Result* of the experiments).

2 Related Works

Several techniques can be applied to extract important sentences. The most common and basic ones are based on the frequency of word appearance [3,7,8]; the position of the sentence in the original paper [8]; and a combination of keyword, location, word instructions and title sentences, along with manual determination of the weight of words [9]. Other studies [4,10] have proposed machine-learning-based approaches to determine the weight of each word automatically. Another technique involves similarity in the contents of sentences (argumentative zoning), such as objectives, problems, conclusion and research results [7,11].

Various features have been implemented to improve the accuracy of the classification techniques. The lexical feature is the simplest and most commonly used [7,12],s because it is relatively simple in its implementation. Researchers have developed lexical features by tokenization of sentences. Tokenization is the process of parsing a sentence or paragraph into units of words, referred to as tokens [3,13]. Each token will be a feature in the classification process [2]. This feature has a positive effect on accuracy.

After these statistical features had been developed, researchers started looking at other possibilities. The concept of natural language processing has gained interest from investigators [14,15]. The researchers began with investigating related features of human language. These features consider the meaning of a

word, such as its semantic features, and relations between words and sentences. An easy way to determine the relationship between sentences is classifying them based on rhetorical categories [2,16]. Teufel, *et al*. [17] introduced 15 rhetorical categories in scientific papers, as shown in Table 1.

Table 1 List of 15 Rhetorical Sentence Categories Developed by Teufel [17].

Category	Description
AIM	Statement of specific research goal, or hypothesis of current paper
NOV_ADV	Novelty or advantage of own approach
CO_GRO	No knowledge claim is raised (or knowledge claim not significant for the paper)
OTHR	Significant knowledge claim held by somebody else. Neutral description
PREV_OWN	Significant knowledge claim held by authors in a previous paper. Neutral description
OWN_MTHD	New knowledge claim, own work: methods
OWN_FAIL	A solution/method/experiment in the paper that did not work
OWN_RES	Measurable/objective outcome of own work
OWN_CONC	Findings, conclusions (non-measurable) of own work
CODI	Comparison, contrast, difference to other solution (neutral)
GAP_WEAK	Lack of solution in field, problem with other solutions
ANTISUPP	Clash with somebody else's results or theory; superiority of own work
SUPPORT	Other work supports current work or is supported by current work
USE	Other work is used in own work
FUT	Statements/suggestions about future work (own or general)
TEXTUAL	Indication of paper's textual structure.

Some researchers classify rhetorical sentences only in the abstract of a paper. The rhetorical structure of the abstract is defined by the order of solving the proposed problem according to five categories of rhetorical sentences, i.e. *Background*, *Objectives*, *Methods*, *Results*, and *Conclusions* [7,15]. We investigated the spread of rhetorical sentences in the abstract and five section classes in the entire paper. We found that rhetorical sentences appear more often in their matching sections, as shown in Table 2. The occurrence frequency of the Result class in the *Abstract* was only 17.7%, while it was 55.5% in the *Experiment Result* section. Hence, extraction of rhetorical sentences is better done on the full text of a paper.

Table 2 Occurrence Frequency of Rhetorical Sentences According to Section Class [10].

	Abstract	Intro	Rel. work	Method	Exp. Result	Conclusion	Reference
Problem	42.1 %	42.1 %	4.7 %	5.2%	5.2%	-	-
Data	33.3 %	7.1 %	6.3 %	16,6 %	26.2 %	9.52 %	-
Method	21.2 %	14.9 %	2.2 %	27.6%	12.7 %	14.9 %	-
Result	17.7 %	2.2 %	-	-	55.5 %	22.2 %	-

Different from Teufel, our method uses only five section classes and classified them automatically using unsupervised learning. We propose to use these five section classes because the format of experimental technical papers is typical. A standard rhetorical division has been established, consisting of *Introduction, Proposed Solution, Experimental Design, Result, Discussion,* and *Conclusion.*

We reduced the 15 types of rhetorical sentences from Teufel to 4 because our corpus is a specific type of scientific paper. Teufel used general scientific papers, whereas we used technical-experimental scientific papers. This type of paper has 4 specific types of rhetorical sentences, i.e. *Problem, Method, Data* and *Result*. However, we also used *Title* as a feature. We added the amount of data, re-evaluated the preposition patterns and used a pattern-matching method for the classification process. This finding was the basis for the idea of adding a section class feature.

3 Our Method

In the present study, we implemented a new technique to extract the **sectionClass** feature and the **Title** feature. The **Title** feature is related to the title of the paper. We investigated a number of titles of papers and found that almost all titles contain at least one preposition. A preposition separates rhetorical words or phrases. Each phrase has a different meaning. We utilized the preposition patterns that appear in titles to find out the meaning of a rhetorical word or phrase. The data set contained 744 titles. We collected 56 preposition patterns from the data set and used these patterns for automatic classification of rhetorical phrases. It was found that both features (**sectionClass** and **Title**) improved the accuracy value.

This study focused on the classification of rhetorical sentences in four categories: *Problem*, *Method*, *Data* and *Result*. Some examples of rhetorical sentences for each category are shown in Table 3. One cue that rhetorical sentences have in common is a special word that is often used when authors convey their intent. This word can be used as a feature for identifying the appropriate category. The name of this feature is **indicativeWord**. This study also investigated sections where rhetorical sentences most often appear. For example, rhetorical sentences about the method used often appear in the *Method* section and rhetorical sentences about the results are commonly seen in the *Conclusion* section. Thus, sections can be used as a feature, which is called **sectionClass**.

To achieve a gold-standard performance of classification, this study also implemented the **Title** feature. As mentioned before, this feature comes from the title of the paper [18,19]. The result of combining these features is better

precision. In other words, this study proposes the best techniques and features to improve the precision of rhetorical sentence classification.

Table 3 Examples of Rhetorical Sentences for Problem, Method, Data, and Results.

Category	Sentences
Problem	*1. The recognition and storage of complex relations among subjects mentioned in these sources is a very difficult and time consuming task, ultimately based on pools of experts.* *2. Hence text mining techniques based on pure linguistic strategies fail to extract information from texts.*
Data	*1.3.1 Experimental Set-up The experimental corpus, made of 86 documents, annotated by two teams of analysts, has been extracted from two collections of public judicial acts related to the legal proceedings against the same large criminal enterprise.*
Method	*1. SVMs here are employed to produce a set of possible interpretations for domain relevant concepts.* *2. The common idea of these works is that the computation of the function f (as in Eq. 1) is translated into an automatic classification step.* *3. SVM classifiers learn a decision boundary between two data classes that maximizes the minimum distance, or margin, of the training points of each class from the boundary.*
Result	*1. The empirical investigation presented here shows that accurate results, comparable to the expert teams, can be achieved, and parameterization allows to fine tune the system behavior for fitting the specific domain requirements.* *2. Although the precision score of Decision Tree and Naïve Bayes are better than the model trained over the bag-of-words (i.e. a simple model), it achieves an overall lower F1 measure (0.31 and 0.4 vs. 0.45) this is due to the higher generalization power of the kernel methods, in fact the simple Bow model is already able to achieve an higher recall level.* *3. On some more complex relationship classes, as PP knows PP and PP hangs out at a Pl, the KXBOW kernel achieves lower performances, basically due to the presence of dialectal or syntactically odd expressions.*

The tasks of classification in this paper are: (1) section classification for building the **sectionClass** feature, (2) title extraction for building the **isContainsTitle** feature, (3) automatic rhetorical sentence classification, which is the end goal of this study. These tasks are shown in Figure 1.

3.1 Automatic Section Class Classification

In other studies [1,2,15], there is no explanation of how sections were identified and classified. Our method uses Machine Learning to classify sections. There are three steps in the classification process: (1) divide the body of the paper into four zones, (2) classify the content of the paper's zones into two classes, and (3) classify the sections into five section classes. The input of this task are experimental papers and the output are the same papers classified according to the section classes.

3.1.1 Divide the Body of the Paper into Four Zones

This is a new proposition of this study. In this process, we divide the body of the paper into four general zones. We divide the text into four zones because we need to separate the sections from each other and then we extract the sections based on their class.

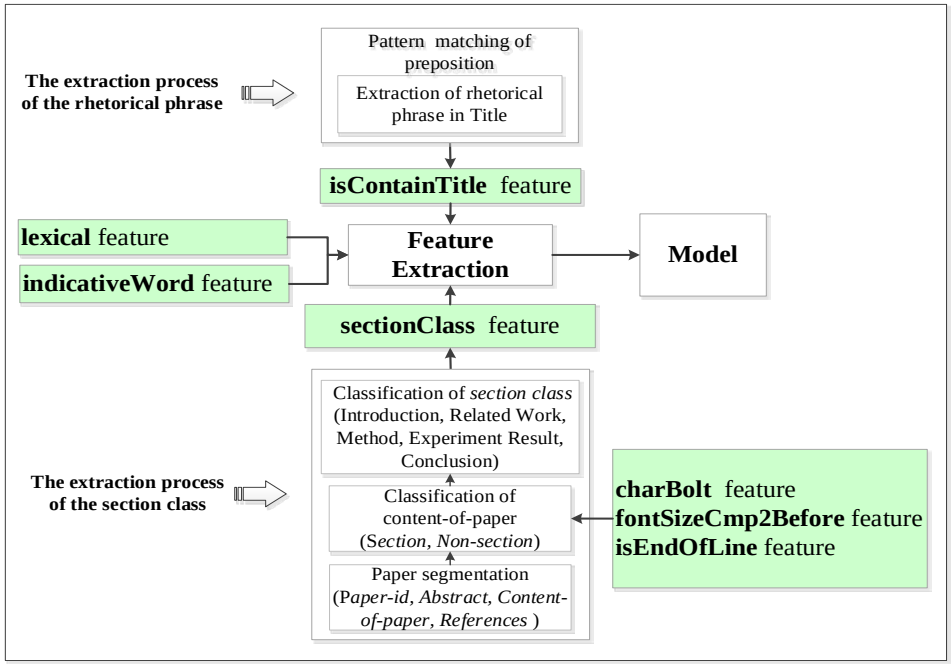

Figure 1 The Extraction processes of the **sectionClass** and **isContainTitle** features.

The sequence of the zones is: *Paper-id* zone at the beginning of the paper, *Abstract* zone, *Content-of-paper* zone, and *References* zone. Our method uses the regular expression method because this method is suitable for segmentation with definite features. The technique is as follows:

The *Paper-id* zone starts with the initial character of the paper and continues until the word “Abstract” is found.

The *Abstract* zone starts from the word “Abstract” and continues until the word “Introduction” is found. The regEx pattern for the abstract zone is:

regexAbstract = "(a[ ]*b[ ]*s[ ]*t[ ]*r[ ]*a[ ]*c[ ]*t).

The *Content-of-paper* zone starts from the word “Introduction” and continues until the word “References” is found.

The regEx pattern is:

regexIntro = "[\n](" + regexPrefixMix + ")[ ]*"+ "((i[ ]*n[ ]*t[ ]*r[ ]*o[ ]* d[ ]*u[ ]*c[ ]*t[ ]*i[ ]*o[ ]*n)|" + "(o[ ]*v [ ]*e[ ]*r[ ]*v[ ]*i[ ]*e[ ]*w))".

The last zone is the *Reference* zone. This zone starts from the word “References” and continues until the end of paper.

3.1.2 Section Classification

Each line in the *Content of paper* zone is classified into two classes: Section and Non-Section, using the Naïve Bayesian algorithm. The number of sections is no more than 10, while the number of lines in a paper is approximately 300.

Table 4 Features for Line Classification.

Features	Feature value
isBoldChar	*true, false*
fontSizeCmp2Before	*true, false*
isPapragraphStart	*true, false*
isStartWithNumerical	*true, false*
isCapital	*true, fal se*
isEndOfLine	*true, false*

We used two corpuses, an XML corpus and a PDF corpus. Both corpuses were needed to test new features. These new features are based on the format type of sections. We consider that humans easier recognize the format of a section. Three new features were implemented: **BoldChar**, **fontSizeCmp2Before** and **EndOfLine**, as shown in Table 4. These features give better precision. The three new features can be described as follows:

isBoldChar feature: This feature worked only on a PDF corpus, because bold characters cannot be recognized as such in XML files. The PDF corpus consisted of the original files in PDF format. The XML corpus consisted of the

files in XML format. In this study we tried to find a way to synchronize both corpuses. When the name of the file is read in the tag "gate.SourceURL" in the XML corpus, then the system looks for the same file name in the PDF corpus. After that, each line of the PDF file is checked using the **isBoldChar** feature. The value of this feature will be *true* if all characters in a line are bold.

fontSizeCmp2Before feature: This feature also worked only on the PDF corpus. Usually, the size of the characters in a section name is different from the other characters in a paper. This feature compares the characters in a line with those in the two previous lines. The reason for this is that the two previous lines belong to the body of the paper, which has normal-size characters. It is not possible to use this feature by comparing the first previous line because that is a blank line. This feature will be marked as *true* when the size of the characters is bigger than that in the previous two lines.

isPapragraphStart feature: The first line of a section is definitely located in a new paragraph. New paragraphs are marked by a blank line or <enter> before the new paragraph. The value of this feature is *true* when there are two <enter> characters before a line.

isStartWithNumerical feature: In the standard scientific paper format, a section begins with a numerical value, a roman numeral or a single character. This feature detects whether a line begins with a numerical value, a roman numeral or a single character. The value of this feature is *true* if the line has one of the three types of characters.

isCapital feature: Each word of the first line of a section always begins with a capital letter. The value of this feature is *true* if a word begins with a capital letter.

endOfLine feature: This feature will be *true* if the end of a line has the <enter> character and does not have the <.> character.

This classification produces two classes: (1) the first class contains lines that are identified as the title of a section. We call this the Section class; (2) the second class contains lines that are identified as not the title of a section, so we call this the Non-Section class.

3.1.3 Classifying the Section Class into Five Section Classes

Each line in the Section class is classified into five classes: *Introduction, Methods, Related-Work, Experiment-Result,* and *Conclusion*. This classification

is done to mark the section classes where rhetorical sentences appear. Then, the Section class is used for feature extraction.

Table 5 List of Features to Classify any Section Class Instance into Five Classes (*Introduction, Related work, Method, Experiment-Result,* or *Conclusion*).

Features	Feature value
isFirstSentence	*true, false*
lengthNumeric	*Numeric*
sectionPosition	*Numeric*
isCapital	*true, false*
isPrefixType	*true, false*
isNewParagraph	*true, false*
wordContained	*word/ phrase*

Some of the features of section classification are reused in this process with the intention to filter out the lines in a class that do not belong in this class. The features used in the rhetorical sentence classification process are shown in Table 5. The description of these features is as follows:

isFirstSentence feature: This feature represents the sentence position in the paper. We have explained this in our previous research.

lengthNumeric feature: In general, the number of characters in a section line is not larger than the maximum number of characters in one line. This feature value is a numerical according to the number of characters contained in a line.

sectionPosition feature: This feature determines the relative position of a section in the full paper. The value of this feature is the position of the initial character of the section divided by the total number of characters in the paper.

isCapital feature: This feature is also used in an earlier phase. It determines whether each word in a line begins with a capital letter. If so, the **isCapital** feature is *true*.

isPrefixType feature: This feature has been described in the first phase.

isNewParagraph feature: This feature has been described in first phase.

wordContained feature: This feature indicates whether a line includes a clue word. Here, we list several clue words that usually exist in a specific section, such as *Abstract, Introduction, Relation Work, Method, Experiment Result, Conclusion, References.*

3.2 Preposition Pattern Matching for Rhetorical Phrases in Titles

Then we also extracted rhetorical phrases from the title as a feature to classify rhetorical sentences. From our previous study [19], we adopted that the title of a paper contains information about *Problem* (P), *Method* (M), and *Data* (D). We assumed that *Problem* and *Task* are the same in experimental research. *Method* is the proposed way to solve the problem and *Data* is the object of the research. These phrases are separated by prepositions. To determine the class of a phrase, we employed two annotators, who have the same research field. The results were evaluated together with our supervisors. We analyzed 744 titles of papers and found 56 preposition patterns that appear more than three times, and discovered that the title always contains at least the information of the research problem. The number and sequence of prepositions determine the label of the phrase, as in the following examples:

1. <M> Statistical Models <M> *for* <P> Text Segmentation <P>
2. <P> Topic Tiling <P> : <M> A Text Segmentation Algorithm <M> *based on* <M> LDA <M>

The first example contains one preposition, {... *for* ...}. The phrase before the '*for*' preposition is the *Method* of the research while after the *for* preposition is the *Problem*. The second example has two prepositions {.. : ..based on..}. Before the ':' preposition is the *Problem*, after the ':' preposition is the *Method*, and after the '*based on'* preposition is the *Method*.

Table 6 Patterns of Prepositions that Appear More Than Three Times.

Preposition	Pattern of Preposition
NON	P
ONE	P - based on - M, P - by - M, P – with - M, P - using - M, P - via - M, P - from - D, P - in - D, M - to -P, M - for - P, D - : - P
TWO	D - : - M - for - P, D - for - D - : - M, D -for - P - based on - M, D - for - P – using - M, D- using - M – for - P, D - using - M - for - P, M - for - P - in - D, M - for - D - in - P, M - for - D - using - M, M - for - P - by - M, M - for - P - for - D, M - for - P - in - D, M - for - P - on - D, M - for - P - using - M, M - for - P - with - D, M - from - D - for - P, M - from - D (M - for - P), M - from - D - for – P, M - for- P - using - M, M - in - D - for - P, M - in - P - : - P, M - in - P - for - D, M - in - P - in - D M - to - P - for - D, M - to - P - in - D, M - to - P - using - M, M - to - P - using - M, M - with - D - for – P, M - with - M - for - D, P - based on - M - for - D, P - for - D - using - M, P - in - D - : - M, P - in - D - using - M, P - using - M - in - D, P - using - M - in - P, P - with - M - from - D, using - M - for - P, using - M - to - P
THREE	M - to - P - with - M - from - D, M - for - P - with - M - to - P,

Preposition	Pattern of Preposition
	M - for - P - from - D - to - P, P - using - M - from - P - to - P, M - : - M - to - P - from - D, M - for - P - using - M - in - D, M - for - P - using - M - in - D, P - using - M - in - D - using - M, using - M - for - P - in - D, using - M - to - P - from - D, using - M - to - P - in - D

The conclusions of these observations are: (1) the maximum number of prepositions in a title is 3 (three) and the minimum is null (without preposition); (2) the preposition patterns vary – we found 59 patterns, as shown in Table 6; (3) the most interesting finding of these observations is that both of the examples have information extraction phrases, but they are labeled differently. The first one is *Method* and the second one is *Problem*. The reason for their different labels is that they have different prepositions following them. Therefore, the labels of both phrases are also different. Therefore, the conclusion is that the meaning of a phrase depends on its preposition.

First, the *Problem*, *Method* and *Data* phrases are extracted based on their preposition pattern. Next, the phrase is cleaned from its prepositions, articles and punctuation using Stopwords. The remaining phrase is compared to a sentence. If the sentence contains one of the phrases (e.g. *Problem*), the value of the feature **isContainTitleProblem** is *true*.

4 Rhetorical Sentence Classification

After the **sectionClass** and **isContainTitle** features have been completed, the next task is rhetorical sentence classification. This study proposes four categories of rhetorical sentences so that all the sentences in a full paper are classified into these four categories. This process uses the XML corpus whose sections are classified into five section classes, after which the rhetorical sentences in each section are identified. Because this classification has a specific goal, the features used are also specific. Table 7 shows the features used in this classification. The features can be described as follows:

Table 7 List of Feature to Classify any Sentence into Four Rhetorical Classes (*Problem*, *Method*, *Data*, and *Result*).

Group	Features	Value
Section	*sectionClass*	*{Abstract, Introduction, Related Work, Method, Experiment Result, Conclusion and References}*
Lexical	*Lexical*	*Numeric*
Title	*isContainTitleProblem*	*true/ false*
	isContainTitleMethod,	*true/ false*
	isContainitleData	*true/ flase*

Group	Features	Value
IndicativeWord	*indicativeWordProblem*	*true/ false*
	indicativeWordData,	*true/ false*
	indicativeWordMethod,	*true/ false*
	indicativeWordResult	*true/ false*

sectionClass feature: This feature is a result of the first phase and identifies the section class in which rhetorical sentences may appear. The feature value is one of the section classes, i.e *Abstract, Introduction, Related-Work, Method, Experiment-Result,* or *Conclusion*.

lexical feature: The lexical feature is a common feature implemented in this type of classification. This feature works by individually checking words based on their grammatical aspect. A lexical unit consists of a word and its grammatical aspect, for example pronoun, verb, noun, etc. This feature uses the *term frequency–inverse document frequency* concept and then calculates the feature vector for each word.

Table 8 Collection of Indicative Words.

Problem	Method	Data	Result
Problem	we proposed	learning corpus	the result
we focus	in this paper	we used	precision better
we discuss	we used	the corpus	than
	we take	web corpus	recall better than
	our method	the data for	result the best
	we considered	wikipedia	achieve
	this paper proposes	manually	we have proved
	this paper also explores	annotate	value
	this article	the ace 2004	summary
	we implement	we used ace 2004	quite well
	our dynamic	we took ace 2004	evaluation
	we can take	training data	
	our work	data source	
	articles are processed	instance	
	we automatically		
	we describe		

isContainTitle feature: The feature group **isContainTitle** has three features, i.e. *isContainTitleProblem, isContainTitleMethod,* and *isContainTitleData*. These features have very strong semantic dependency on the title of the paper. When a word or phrase contained in the title is also contained in a sentence, then the value of this feature for the sentence is *true* (according to the title of the label).

indicativeWord feature: This feature group is developed in accordance with the objective of the research. This feature is the result of discussions conducted by our research team to find characteristics of words that can identify rhetorical sentences. The words are kept in a dictionary, as shown in Table 8.

5 Experiment and Result

We performed three experiments: extract the **sectionClass** feature, extract a rhetorical phrase from the title, and classify the rhetorical sentence that is proposed. The number of lines for each classification is shown in Table 9. In this task, our method uses the Naïve Bayesian (NB) algorithm. This provides a simple approach using probabilistic knowledge with two simplifying assumptions: conditional independence of features, and no hidden attributes influence the prediction. The NB model contains the probability of each class and the conditional probability of each attribute value given a class. The classification process uses the model to find a class with maximum probability given an instance.

Table 9 Number of Lines for Each Classification.

Classification	Number of instances
Stage I (section – nonSection)	28.304 lines
Stage II (five section classes)	365 lines
Stage III (rhetorical sentences)	xx sentences

5.1 Section – nonSection classifier

The first experiment classified lines into two labels: Section and Non-Section. Then, the experiment was evaluated using F-measure. Learning accuracy achieved 99.45%, i.e. the number of lines classified in the appropriate class was 27.884. The error percentage was 0.65%. Some examples of the errors are shown in Table 10. It is important to investigate the reason why the classification went wrong in different cases.

5.1.1 Section Instances Classified as Non-Section

The first example is “III. EXPERIMENT AND RESULT”. All features worked well on this line. However, the **isCapital** feature value was *false*, which caused the line to go into the wrong class. The task of the **isCapital** feature is to check whether the first letter in a line is capital. We tried to change the definition of the **isCapital** feature, i.e. all the letters in a word have to be in the form of a capital. Then, the adjusted feature was implemented to test each line. However, we still did not get a satisfactory result. Finally, we decided to utilize this feature only to check the first letter.

Table 10 Examples Of Lines That Were Incorrectly Classified.

Non-Section class	Section class
III. EXPERIMENT AND RESULT	A. Data Sets
4. Methods	9 WordNet Semantic Concepts
INTRODUCTION	United Nations Secretary-General
FURTHER RESEARCH	9. Acknowledgements

The second example is "4. Methods". The values of **isStartWithNumerical**, and **isUpperCase** were *true*, but the value of **isCompareTwoLineBefore** was false. This specific example had the same font size as the previous two lines. Similarly, for the **isBold** feature we found that the value was *false* because the line was not in bold characters.

The third example was not only influenced by **isCapital** but also by the **isStartWithNumerical** feature. The feature values of this line were *false*. Also, "FUTURE RESEARCH" was not found in certain papers, therefore this line was considered Non-Section class.

5.1.2 Non-Section Instances Classified as Section

The phrase "A. Data Sets" can be assumed to be in a sub-section of a paper. However, in this specific case the author's writing style was to number these with the numbers 1, 2 and so on. Then to mark sub-sections he used the alphabetic characters A, B and so on. However, the **isStartWithNumeric** feature assumes that this line belongs to the Section class. Thus, the **isStartWithNumeric** feature value was *true*.

Next, the phrase "9 WordNet Semantic Concepts" was not meant to be in Section class. This was actually the continuation of a sentence on the previous line. Hence, the **isCapital** and **isStartWithNumeric** feature values were *true*. The values of the **isCapital** and **isNewParagraph** features in the third line were also *true*, so that this line automatically went into the Section class.

All features worked well in the last example, i.e. the phrase "Acknowledgments", but the ***wordContained*** feature was *false* because there was no class defined to handle this line. Moreover, this line did not have a numeric character and authors rarely write this phrase in their papers. Therefore this study ignores it.

According to the standard evaluation metrics for document retrieval (the precision value, recall, and F-measure), the precision value of the experiment was 0.65, the recall value was 0.95, and the F-measure was 0.77, as shown in

Table 11. Although precision and recall were very different, the F-measure remained high because of the average of precision and recall.

Then, we investigated the three most influential features in this experiment. They were **isBoldChar**, **fontSizeCmp2Before** and **isStartWithNumerical.**

Table 11 Precision, Recall and F-measure using Naïve Bayesian Classifier.

Class	Precision	Recall	F-measure
Section	0.65	0,95	0.77
nonSection	1	1	1

The **isBoldChar** feature was the best-performing one, because one of the characteristics of the section class is that it uses a bold font for class names. Therefore it is very appropriate to implement the **isBoldChar** features in this classification. Moreover, the size of the class name's font was usually larger than the size of the paper's body font. So the **fontSizeCmp2Before** feature also caused better precision.

The **isStartWithNumerical** feature can also be considered as one of the best features, because almost all the section classes began with a numeric value ..., I ..., 1 ..., etc.

Overall it can be concluded that the addition of the three new features considerably contributed to precision, recall and F-measure.

5.2 Classification into five section classes

The second stage of the classification process is to label each line contained in each section class. Each line was classified into five pre-defined section classes. The accuracy of this classification achieved 91.2%. Overall, the precision and recall of this classification achieved good values, causing the F-measure to remain high. The influence of each feature on each section class can be described as follows:

***Introduction* class**: This class is always located at the beginning of a paper. It is always called 'Introduction'. Therefore the **indicativeWord** feature will be *true.* Then, the position of the Introduction class is always located below the Abstract class. Thus, the value of the introduction's relative position is almost the same for every paper. This causes the **sectionPosition** feature to be *true*. Another common cue of this class is that it is always written on a new line. It is marked by a <.> character to terminate the line and an <enter> character to move to the next line. Each of the first letters of the word is a capital letter,

causing the value of the **isNewLine** and **isCapital** feature values to be *true*. Another characteristic is that at the beginning of this section class there is always a blank line. Thus, the value of the **isNewParagraph** feature is *true*. Moreover, it is also common to see a numerical value at the beginning of the name of a class. If an author uses this kind of style (numbering), then the class titles will always have the following pattern: *1. Introduction*, *2. Related Work*, etc.

Related Work class: Some conditions cause the precision and recall of this class to be lower than those of the Introduction class, because this class does not always exist in a paper. Moreover, some authors write this class using a different word or phrase, such as *Background*. Therefore, the value of this feature is not always *true*. In other cases, the position of this class in every paper is not always after the introduction. Some authors put it after the Experiment Result class. Meanwhile, other features work well with this class.

Method class: The precision and recall of this class is higher than in previous results. Adding new features increased the recall value to 0.9, whereas the precision value dropped to 0.85. This value did not affect the F-measure, which remained constant (0.9). In a previous research, any section that was considered to belong to the Method class was merged into one class. This study separated the Method class into sections. Each section is still named Method class. This change resulted in better precision and recall values.

Experiment Result class: The precision and recall value in this class were both 0.91. The average increase was 16%. This proves that this feature was used appropriately and in accordance with the characteristics of this class. The same change as discussed under the Method class was also made in the Experiment Result class, because this class is not always present in one section only, sometimes it is present in two separate sections: the Experiment section and the Result section. Therefore we separated this class into sections. Each section is still named Experiment Result class.

Conclusion class: Same as the previous classes. The increase of the F-measure value for this class was 4%. This is a positive response after adding several new features in the section classification process.

The precision, recall and F-measure of the five section classes are shown in Table 12. Another change we implemented in the classification process was not to include the Abstract class anymore because this class does not have a numeric value. Consequently, the **isPrefixType** feature did not work on the Abstract class. Classification for this class was done separately using the technique of regular expressions.

Table 12 Metric Evaluation for Five Section Classes.

Class	Precision	Recall	F-measure
Introduction	1	0.96	0.98
Related Work	0.94	0.68	0.79
Method	0.85	0.95	0.90
Experiment Result	0.91	0.91	0.91
Conclusion	0.90	0.97	0.94

The best-performing three features in this classification were: **wordContained, sectionPosition**, and **firstSentence**. As for the **wordContained** feature: if the name of a class does not vary (except for Method), then this feature value is always *true*. Those words that can indicate classes that have been saved in our dictionary are quite sufficient to cover all possible words that appear as a class name. **SectionPosition** feature: The relative position of some sections in a paper is quite similar. Only the Related Work class is sometimes placed after the Experiment Result class. The last feature is **firstSentence**. This feature is quite determining in accordance with the section characteristic that they always start with a new line.

5.3 Pattern Preposition in Title

The preposition pattern experiments were performed independently. The data used were 434 titles. Testing data that were part of the training data were cleansed of tags to be used for the separator string and added with other data. The experiment consisted of two parts. The first part was to determine preposition matches without involving following string patterns. The second part was to match the following strings patterns.

The first experiment produced 276 matching patterns and the second produced 95 matching patterns with the same following string. Hence, the total amount of correct matches was 276 + 95 = 371. The accuracy value was 86%. Some of the results and the percentages of each class of phrases in their own class are shown in Table 13. We found Method phrases and Problem phrases difficult to distinguish. Both classes can have the same content. In other words, a phrase in a title could be in the Problem phrase class, but in another title it could be in the Method phrase class. The percentage value of these phrases were 21.3%.

Table 13 Number of Phrase Based on Their Class.

Phrase	Number phrase	%
Problem	438	73.2 %
Method	427	71.4 %
Data	93	15.5 %
Total numbers	598	

The first pattern is correct. The system could identify both rhetorical phrases in accordance with the label. For the second pattern no match was found. This happened because the pattern was not read from the preposition. For the third pattern a match was found. The error occurred because there were two patterns of the same preposition but the label of the string that followed the preposition was different. As shown in Table 14, there were 2 preposition patterns that were the same but had a different label: *M - for - D - using - M* and *M - for - P - using - M*. When pattern matching is performed, the system will use the first pattern found.

Table 14 Some Preposition Patterns after the Matching Process.

No	Output
1.	raw: <m>a seed-driven bottom-up machine learning framework</m> for extracting relations of various complexity clean: a seed-driven bottom-up machine learning framework for extracting relations of various complexity Processed: <M>a seed-driven bottom-up machine learning framework</M> for <P>extracting relations of various complexity</P>
2.	Result: True raw: large scale learning of relation extraction rules with <m>distant supervision</m> from <d>the web</d> clean: large scale learning of relation extraction rules with distant supervision from the web Processed: <P>large scale learning of relation extraction rules</P> with <M>distant supervision from the web</M>
3.	Result: False raw: <m>the role of technology</m> in knowledge management : trends in the Australian corporate environment clean: the role of technology in knowledge management: trends in the Australian corporate environment Processed: <P>the role of technology</P> in <D>knowledge management</D>: <M>trends in the Australian corporate environment</M> Result: False

5.4 Rhetorical Sentence Classification

In our study, rhetorical sentences were classified into four classes: Problem, Method, Data and Result. The number of instances in the experiment was 770 sentences. Four features were implemented in this classification, **sectionClass**, **lexical**, **isContainTitle**, and **indicativeWord**. These four features contributed to increase precision. Like the **lexical** feature, this feature is rich with words, so the **lexical** feature was also utilized in this classification.

The experimental scenario conducted was to test all the features simultaneously and try to create some combination among the four features. To evaluate this experiment we utilized F-measure because F-measure is a representation of the value of precision and recall. The results of the classification and combination process are shown in Table 15.

Table 15 F-measure Value for Combination of Features using Naïve Bayesian Classifier.

Features	F-measure			
	Problem	**Method**	**Data**	**Result**
sectionClass, isContainTitle, indicative Word	0.68	0.71	0.69	0.76
sectionClass, lexical+indicativeWord	0.67	0.71	0.69	0.77
sectionClass, lexical	0.67	0.71	0.69	0.77
sectionClass,indicative	0.61	0.63	0.21	0.71
lexical,indicative	0.56	0.64	0.63	0.69
sectionClass	0.66	0.64	0	0.71
Lexical	0.57	0.64	0.62	0.68
indicativeWord	0	0.31	0.8	0.48

Overall, the F-measure value of the four features was higher than that of the combined features, except for the value of Result, which was 1% lower than the value of the combined feature without using the **isContainTitle** feature. **IsContainTitle** does not work on Result, because titles never contain the rhetorical phrase “result”. The F-measure of Data was 0.69. This was caused by the number of occurrences of Data. This was significantly lower compared to the other rhetorical sentence classes.

Besides the Naïve Bayesian algorithm, this study also tried out the support vector machine (SVM) classification algorithm. We compared SVM with the Naïve Bayesian algorithm to determine which one performed better. The F-measure values using SVM are shown in Table 16. SVM with the combination of all features was slightly superior in classifying rhetoric sentences. The F-measure values increased 1% for the Problem class, 2% for the Method class, 3% for the Data class, and 1% for the Result class.

The **lexical** feature produced similar F-measure values. However, when the combination of all features using SVM was compared with the Naïve Bayesian classifier, the F-measure values tended to rise 1% only for the Problem class. Likewise, the independent **sectionClass**, **lexical** and **indicativeWord** features generally increased by about 2% when using the Naïve Bayesian classifier.

Table 16 F-measure Value for Combination of Features using SVM Classifier.

Features	F-measure			
	Problem	**Method**	**Data**	**Result**
sectionClass, lexical, isContainTitle, indicativeWord	0.69	0.73	0.72	0.77
sectionClass, lexical+indicativeWord	0.69	0.74	0.71	0.77
sectionClass, lexical	0.69	0.74	0.71	0.77
sectionClass,indicativeWord	0.59	0.63	0.14	0.70
lexical,indicativeWord	0.55	0.65	0.65	0.68
sectionClass	0.60	0.63	0	0.71
Lexical	0.55	0.66	0.66	0.68
indicativeWord	0	0.34	0.07	0.44

We looked into classifications of rhetorical sentences into classes that were erroneous. Some examples of rhetorical sentences classified in a class that was not appropriate were:

(The sequence of the features is as follows: sectionClass, isContainTitleData, isContainTitleMethod, isContainTitleProblem, indicativeWordMethod, indicativeWord-Problem, indicativeWordResult, indicativeWordData: Class.)

Example 1: Instance: *"The method is based on information made available by* shallow semantic *parsers."*

Feature value: Abstract, false, true, false, false, false, false, false: METHOD

Predicted class: PROBLEM

Title: Shallow Semantics for Relation Extraction

XML name: docD4

(The value of **inTitleMethod** is *true* and **sectionClass** is Method. This instance should have been included in the *Method* class but it was included in the Problem class).

Example 2: Instance: "1) An Evaluation Method of Feature Selection: We used the cohesiveness of clusters to measure the performance of feature selection methods."

Feature values: (exp.-result, false, true, false, false, false, false, false, false: METHOD)

Predicted class: RESULT

Title: Text Clustering with Feature Selection by Using Statistical Data

XML name: docA2

The value of **sectionClass** was Abstract and the value of **inTitleMethod** was *true*. This instance was supposed to be in the Method class, but the **indicativeWord** "result" (measure and performance) caused the instance to be included in the Result class.

Both examples represent instances that were classified in the wrong class. Almost each sentence that was classified in the wrong class had two categories that occur in the **indicativeWord** dictionary. We inferred that this was the reason why the instances ended up in the wrong class. Another possible cause is that the list of four categories in the **indicativeWord** dictionary is not sufficient.

We compared the result of this research with our previous study and found that our new features increased the performance of classification up to 14% using Support Vector Machine (SVM), as shown in Table 17.

Table 17 Improvement of Classification Performance with Our Previous Research.

Rhetoric class	F-measure (SVM)	
	Last Feature	New Feature
Problem	0.67	0.69 (2.9 %)
Method	0.64	0.73 (14 %)
Data	0.70	0.72 (2.9 %)
Result	0.69	0.77 (11.5 %)

6 Conclusion and Discussion

The **sectionClass** feature worked well and can provide a positive contribution when implemented independently so it can increase the value of the F-measure. However, when it was implemented independently on the rhetorical sentence class Data, the **sectionClass** feature could not identify this sentence as a rhetorical sentence; the value of F-measure was 0 (zero). This was caused by the occurrence percentage of this phrase being far lower than that of other rhetorical sentences, so that this sentence does not dominate in one of the six pre-defined section classes. It also is affected by the number of instances. The occurrence number of this rhetorical sentence is very low compared to that of the other rhetorical sentences. However, when this feature was combined with the **lexical**

feature, the F-measure value could still reach an average level because the **lexical** feature is rich with words.

The precision of the **isContainTitle** feature is low because the assumption of the annotation is still weak. In future research, we will develop a classification for rhetorical phrases, specifying a clear distinction between each phrase.

References

[1] Teufel. S., *Argumentative Zoning: Information Extraction from Scientific Text*, Ph.D Dissertation, University of Edinburgh, Edinburgh, Scotland, 1999.

[2] Kodra, M.L., Widyantoro, D.H., Aziz, E.A. & Trilaksono, B.R., *Information Extraction from Scientific Paper Using Rhetorical Classifier*, International Conference on Electrical Engineering and Informatics, Bandung, Indonesia, 2011.

[3] Edmundson, H.P, *New Methods in Automatic Extracting*, Journal of the ACM (JACM), **16**(2), pp. 264-285, 1969.

[4] Helen, A., Widyantoro, D.H. & Purwarianti, A., *Extraction and Classification of Rhetorical Sentences of Experimental Technical Paper Based on Section Class*, Second International Conference on Information and Comunication Technology (ICoICT), 978-1-4799,3580-2, pp. 419-424, IEEE, 2014.

[5] *APA Experimental Paper Writing Format*, https://owl.english.purdue.edu/owl/resource/560/13/ (16 October 2015).

[6] Helen, A., Purwarianti, A. & Widyantoro, D.H., *Developing the Research Map Framework to Present the Positioning Research Automatically*, Proceeding of Information System Conference (KNSI), pp. 1458-1465 Mataram, Februari, 2013. (Text in Indonesian)

[7] Teufel, S., *Argumentative Classification of Extracted Sentences as a First Step Towards Flexible Abstracting*, In Advances in Automatic Text Summarization books, Mani and M.T. Maybury (Eds.), pp. 155-176, 1999.

[8] Luhn, H.P., *The Automatic Creation of Literature Abstracts*, IBM Journal of Research and Development, **2**(2), pp. 159-165, 1958.

[9] Kupiec. J., Pedersen, J. & Chen, F., *A Trainable Document Summarizer*, Proceeding of ACM SIGIR, pp. 68-72, 1995,

[10] Baxendale, P.B., *Machine-made Index for Technical Literature – An Experiment*, IBM Journal of Research and Development, **2** (4), pp. 354-361, 1958.

[11] Shiyanm. O., Khoo, C.S.G. & Goh, D.H., *Design and Development of A Concept-based Multi Document Summarization System for Research Abstracts*, Journal of Information Science, **34**, pp. 308-326, 2008.

[12] Yamamoto. Y., & Takagi, T., *A Sentence Classification System for Multi Biomedical Literature Summarization*, Proceeding of the 21st International Conference on Data Engineering Workshops, pp 1163 IEEE Computer Society Washington, DC, USA, 2005.

[13] Verma, T., Renu, R. & Gaur, D., *Tokenization and Filtering Process in Rapid Miner*, International Journal of Applied Information Systems, **7**(2), pp. 16-18, 2014.

[14] Futrele, R.P., Satterley, J. & McCorma, T., *A New NLP System for Biomedical Text Analysis*, NLP-NG, IEEE 978-1-4244-5121-0, pp. 296-301, 2009.

[15] Ungurean, C. & Burileanu, D., *An Advanced NLP Framework for High-Quality Text-to-Speech Synthesis*, IEEE 978-1-4577-0441-3, pp. 1-6, 2011.

[16] Raje, S., Tulangekar, T., Waghe, R., Pathak, R. & Mahale, P., *Extraction of Key Phrases from Document using Statistical and Linguistic Analysis*, Proceedings of 4th International Conference on Computer Science & Education, IEEE 978-1-4244-3521-0, pp. 161-164, 2009.

[17] Teufel, S., Siddhartan, A. & Batchelor, C., *Towards Discipline-Independent Argumentative Zoning Evidence from Chemistry and Computational Linguistics*, Singapore, Proc. of the Conference on Empirical Methods in Natural Language Processing, 2009.

[18] Yu, F., Xuan, H.W. & Zeng, D., *Key-Phrase Extraction Based on a Combination of CRF Model with Document Structure*, Eighth International Conference on Computational Intelligence and Security, IEEE 978-0-7695-4896-8, pp. 406-410, 2012.

[19] Widyantoro, D.H. & Helen, A., *Preposition-based Pattern Sequence for Rhetorical Phrase Extraction in Title Scientific Papers*, in Proceeding RCCIE (Regional Conference on Computer and Information Engineering), Yogyakarta 7-8 October, 2014.