Automatic Identification of Compare Paper Relations

Yuliant Sibaroni, Dwi Hendratmo Widyantoro, and Masayu Leylia Khodra

School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
yuliant@telkomuniversity.ac.id, dwi@stei.itb.ac.id, masayu@informatika.org

Abstract: One important issue in performing a good research is to *compare* their current research with results of others. However, the comparison requires two or more papers to form a so-called *compare-paper* relation. This comparison can be identified based on their citation context, that consist of high number of sentences, rather than through basic features such as N-Gram. To investigate the relationship between papers, Wang et al. and several other researchers have applied the cue phrase feature, which was obtained via a manual analysis of a data set of scientific paper. Furthermore, a more complex feature was proposed by Park and Black for the same purpose. Nevertheless, they were unable to investigate accurately such relation, since their features are not made specifically for this purpose. In this paper, we propose new features that specifically intended to identify the relationship of papers or *compare-paper* relation. The experimental results show that the proposed features result in much better performance compared to the experiments by using the best baseline feature. By using 6 different classifiers, the experimental results also show that maximal values result in best values for each classifier. Moreover, other experimental results show that the best performance is obtained by combining the baseline features and the newly developed features, which shows that they are mutually reinforcing.

Keywords: paper relation, compare, feature, cue phrase, text classification

1. Introduction

Visualization of citation networks is a field of research that is currently developing and has the potential to be used in large websites such as Google. In the field of network citation, Eck and Waltman [1] studies a visualization of network citation, where a large number of papers are linked to one another by citation and forms a citation network. The visualization if this citation network is very useful for researchers, especially to see relationship between a research with others.

Along with the rapid development of technology, this citation network is considered as insufficient for researchers. They need a more detailed form of citation; one example is a citation that contains a comparison form. The existence of this relation is very important for research identification. By knowing such relations in a specific research field, it will be possible to find out the state of the art of the paper, and the best method or algorithm that have been used. This relation is essential to produce good research results with significant contribution.

Comparison of performance, complexity, and processing time are components that frequently found in scientific papers. Therefore, sentences that contain a comparison between these entities are referred to as comparative sentences [2], [3]. A comparative sentence, that containing a comparison between entities originating from different papers, indicates a connection between those papers. The paper relation that comprises such comparison is referred as the *compare* relation [4]. This relation identification can be categorized as a text classification problem, i.e. by viewing it as a comparative and non-comparative sentence classification. In the text classification, the accuracy of the comparative sentence classification

Received: February 4th, 2019. Accepted: March 30th, 2020
DOI: 10.15676/ijeei.2020.12.1.12

depends on several factors or processes, such as pre-processing stages, features that are used, feature selection methods, and classifiers.

Among these influential factors, the use of features in the classification process is a major concern of this study, since they are considered to be the most important factor in determining accuracy. This process is analogous with research in the field of bioinformatics; determining the identity of a human being through unique features such as patterns of fingerprint, retina, iris, etc [5].

In identifying the comparative relation between papers, Wang et al. [4] used the *cue phrase* feature and citation to classify the *compare* relation sentences according to the rule-based approach. However, the use of such a relatively simple feature makes the result less accurate since the comparative and non-comparative sentences can sometimes contain the same phrases. A better approach is proposed by Park and Blake [6] who uses more complete features by grouping them into two groups, i.e. Lexicon and Syntax. Nevertheless, they can only be used to recognize comparative sentences in general; not specifically developed to investigate *compare relation* sentences, which result in less accurate results in its application for classification. This can be caused by the fact that the use of Park and Blake's feature could potentially produce many sentences with non-relation's *compare* that are classified as *compare* relation. Note that, in a *compare relation*, there should be a relation between at least two papers.

In this study, we propose new features that can identify the *compare* relation more accurately, i.e. by using sentences that containing a comparison between two or more papers, or in this paper is referred as *compare* sentence. The comparison between two or more papers can be categorized as neutral, in which no paper is superior than others, or non-neutral, in which one research is more superior compared to the other research.

Determining sentences as *compare* sentences is based on the definition of *compare* and *improve* relations that are defined by Wang et.al [4]. As defined [4], a sentence is categorized as a *compare* sentence if it contains "approach or result in one work is better than in another work", where this *compare relation* shows a comparison without comment. In this paper, the process of identifying *compare relations* is done by classifying *compare* sentences. The classification of *compare* sentence is performed by using a machine learning-based approach with the features that are proposed by Park and Blake [6] and the features of Wang et al. [4] as the baseline. The proposed features in this paper combine both Park's and Wang's features while the others were developed based on the baseline along with a series of certain automatic steps. Therefore, the main contribution of this research to existing studies is to produce new stronger features for the process of identifying *compare* paper relations and provide automated procedures for the feature extraction process.

The research conducted in this study, including the results of its contribution are illustrated in Figure 1. Figure 1 shows the proposed research position compared to other studies. This research not only studies a special part of paper relation identification research, but also in other research area, i.e. citation context classification and citation identification. These studies specifically also contribute to supporting research in the field of citation network visualization, namely in the development of visualization of citation network relations. A citation network relation is a more detailed form of a citation network, i.e. a citation between papers that contains a paper relation, especially a *compare relation*.

The organization of this paper is as follows. Chapter II describes studies on papers relation in general, *compare relation*, and *compare* type sentence. Description of important features that are used by the baseline in classifying *compare relation* and *compare* type sentence are described in Chapter III. In Chapter IV, the proposed features along with the process to automatically develop new features are presented. Chapter V discusses the experimental results including the classification performance of *compare relation* sentences using both the baseline and the proposed new features. Finally, the paper is concluded in the final chapter

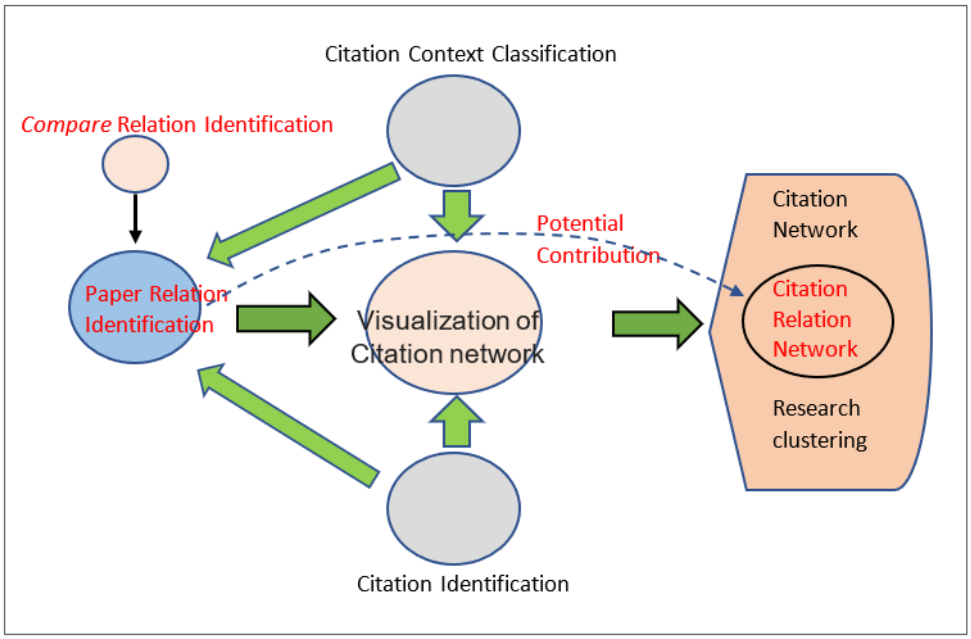

Figure 1. Position of Proposed Research

2. Related Work

A relation between papers is formed when a paper is cited by one or several others because of various reasons. In the research on visualization of citation networks, relations between scientific papers are displayed in graph form where nodes and edges represent paper and citation, respectively [1]. This research is important especially for supporting researcher to find the position of their research with others. When looking at the relationship between two papers based on citation, users usually want to find out more details about the paper relationship. In the citation context, author may make use of data or theorems, discuss weaknesses, or *compare* his or her own research results with others' papers or research.

Research on citation sentence has been studied by various researches, but the focus of this study can be grouped into two parts, namely the identification of citation sentences and the classification of citation sentence. In identification of citation sentences, Widyantoro and Amin [7] employed textual and numbered citation styles, while Sugiyama applied *unigram, bigram, proper Noun, previous and next sentence, orthographic* and *position, and Teufel and Athar used sentence features related to the properties of i-th sentence (Si)* [8].

In the classification of citation sentence, Teufel et al. [9] classified citation sentences into 12 categories. Four of these 12 categories, i.e. CoCoGM, CoCo-, CoCoR0, and CoCoXY, are categories that related with *compare relation*. Here CoCoGM denotes the contrast between methods that are employed or goals, CoCo- denotes the difference between the cited results is worse than the current work, and CoCoR0 denotes the cited results are better or comparable with the current work. Lastly, CoCoXY denotes the citations that are being explicitly compared and contrasted with other work that are not in the current work. These four categories are similar in relation to compare, but these categories do not guarantee the relation between papers in it. In addition, the data that are used in their study is limited to citation sentences. Other researchers involved in this research area are Angrosh et al. [10]. They propose 7 categories, where 2 categories, i.e. CCW and RWCW, are categories related with *compare relation*. Teufel et al. [9] applied an automatic classification by using a cue phrase feature as the main feature, while Angrosh et al. [10] did not and emphasized on the concept of data labelling.

Currently, the only research that focuses and explores paper relations based on citation contexts is the research conducted by Wang et.al [4]. They categorize papers relation based on the content contained in the citation sentence into 4 types, i.e. *extend*, *criticize*, *compare*, and *improve*. To identify these types, they employ a rule-based approach with the cue phrase as its main feature. This is possible since each of them has a different cue phrase that are obtained manually. Here, the cue phrase, that is obtained manually, is based on observations of the sentences that contain the *compare relation*.

With respect to the *compare* relation, only the research of Wang et al. [4] that have been able to define it explicitly by defining its special form as the *compare relation*. This means that a sentence citation will be classified into a *compare* or *improve* category if it contains an explicit comparison between two studies. This is what distinguishes Wang's classification from others. For example, a citation sentence: "Method A is better than B [11]" is not categorized as a *compare* sentence by Wang et al., but according to the rules defined by Teufel or Angrosh et al.[10], it will be categorized as a *compare* or *improve* sentence. The concept of labelling data from Angrosh et al. [10] to this *compare* relation still needs to be implemented automatically to find out the level of effectiveness.

A feature is the main element that greatly determines the accuracy of classification results both for texts and other fields. The use of cue phrase feature in the *compare relation* classification based on the citation sentence proposed by Wang et al. is less optimal. A more complete feature for the *compare* sentence classification was proposed by Park and Blake[6]. Although their research does not focus on the relation sentence, it has a potential for identifying *compare relation* sentences in more detail. In general, these features are a combination of Lexical and Syntax features. Hence, in addition to paying attention to Lexical features as used by Wang et al., Park and Blake also analyse a *compare* sentence structure based on its language structure.

In connection with these studies, the research in this paper aims to complement research related to comparative relations where the main focus of this study is to produce stronger features and provide a procedure for extracting them automatically. To that aim, we combine features created by Wang et al. [4] with features of Park & Blade [6].

3. Basic Feature of *Compare Relation*

In general, two feature groups are used as the baseline for this study. The first feature is derived from Wang et al.'s [4] which consists of *compare* cue phrase features, *improve* cue phrase, and *citation* to identify *compare relation*. The data that are used by Wang et al. was obtained from 40 research papers, i.e. selected randomly from IEEE Transactions that published by Computer Society Digital Library [4]. The second feature is the one that developed by Park and Blake [6], i.e. the feature which was used to classify *compare* sentences.

A. Wang et.al. feature

The feature of Wang et al. consists cue phrases of *compare relation*. Moreover, it become more specific cue phrases when comparing a paper with more superior paper. A sentence will be classified as a *compare relation* when it has a citation and contains one of *compare* or *improve* cue phrases as presented in Table 1. In general, it has 3 features including citation, *cue phrase of compare*, and *cue phrase of improve*. *Cue phrase of compare* is a collection of phrases that identify the existence of a *compare relation*. The definition of *improve* cue phrase is similar to that. These cue phrase give a unique characteristic of a sentence with the *compare* category. These cue phrases are obtained by observing sentences in *compare* sentence manually. For the example, in sentence [4]: "`Compared to [7] and [35], this paper provides random accessible mesh compression with better compression ratio and explicit control of random accessibility`", the sentence above indicates the citation sentence, because there is a citation mark: "[7] and [35]" and there is also a cue phrase that characterizes that the sentence is the *compare* sentence which is "*compare* to".

Table 1. Wang's cue phrase

cue phrase of compare	*cue phrase of improve*
different from, agreement with, compared to, like, similar to, in contrast to, unlike, identical to	*improvement, enhancement, better than, to avoid this problem, to solve this problem*

In Table 1, there are collections of phrases in *compare* cue phrases such as: *different from, agreement with, compared to*, etc., which state a neutral comparison, because it seems that no one party is superior to another. As for the collection of phrases in *improve* cue phrase, it appears that the use of phrases: *improvement, enhancement, better than others* shows that one party is superior than the other party. Indeed, the cue phrase obtained can be different when the corpus used is also different.

B. Park and Blake feature

Park's features used in this study consisted of 35 original features which were made of 6 Lexical, 27 Syntax, and 2 additional features plus 1 citation feature. The additional ones consisted of plural and preposition features, while the citation feature was added to catch relation sentences. The features extraction process was conducted as closely as possible with what was conducted by Park, although some of them required adjustments generally relating to changes in the *Stanford typed dependencies* because usage of different Java library versions. For example, the type of dependency *pcomp* in the old version changes to *ncmod* in the new version [11]. Several others were made since the reference data sources used in Park and Blake's paper were difficult to obtain. Some adjustments to the feature extraction process performed in this paper can be seen in Table 2.

Table 2. Adjustment of Java Library and Others in the Feature Extraction Process

Nbr.	Feature Extraction Process in Park et.al.[6]	Feature Extraction Process in this paper
1	partmod	nfincl [11]
2	infmod	nfincl [11]
3	pcomp	ncmod [11]
4	pobj	nmod [11]

Terms in Table 2 can be explained as follows: nfincl stands for "infinitive nonfinite clause", which is a general form of partmod and infmod. The other term is ncmod which stands for "nominalized clause modifier" which is a transformation of pcomp, while the last term is nmod which stands for "nominalized clause modifier" which is a transformation of pobj.

C. The Combination of Park's and Wang's features

The baseline used in this study is a combination of Wang et al.'s [4] and Park and Blake's features [6]. This combination was based on the reason that the feature used by Wang et al. [4] in identifying process of *compare relation* is still relatively simple, while the features used by Park and Blake appears to be quite complex but not specifically aimed for relation sentences. This combination was expected to produce a better performance than rather than using the two features separately. Thus, there were 38 features in total from which 3 were taken from Wang et al.'s and 35 from Park and Blake's.

4. Proposed Features

The development of new features was performed by recognizing them based on the frequency of their appearance in a *compare* positive class. The high frequency features is potential to identify *compare* sentences accurately. In general, the procedure is as shown in Figure 2.

A. ProportionWord Feature

The ProportionWord feature is a binary word feature that contains a collection of words with high word proportion values in *compare* class. The proportion value of the word A in the *compare* sentence is calculated based on the comparison between the number of *compare* sentences that contain the word against the total *compare* sentence. Implementation of this feature extraction procedure produces a collection of word features as contained in the Table 3.

Table 3. List of Words in ProportionWord Feature

Nb.	Word	Nb.	Word
1	with	12	it
2	our	13	systems
3	system	14	better
4	we	15	which
5	compared	16	other
6	from	17	model
7	by	18	can
8	’s	19	et
9	similar	20	results
10	than	21	parser
11	approach	22	not

The words in Table 3 have a proportionWord value> 10%. This value is calculated as shown in formula 1. A value of 10% is actually classified as a low proportion value. But based on the results of observations made, it is found that the number of words with high proportion values (above 50%) is very small, so that the lower limit of the proportion used is determined to be 10% in order to capture more words. Another consideration is that the proportion of *compare* sentences in the dataset is very small, which is less than 1%, so it will be quite difficult to explore the value of a large proportion. For the dataset, the use of this feature can capture 22 words as seen in Table 3 with proportion values sorted by the highest.

$$\frac{\#\textit{compare sentence contain word A}}{\#\textit{compare sentence}} \quad (1)$$

Considering the small proportion of the *compare* class with respect to the non-compare, the number of occurrences of a feature in the *compare* class is highly considered to be developed into the new one. Detailed algorithm to extract the *ProportionWord* feature can bee seen in Figure 2.

The ProportionWord feature extraction algorithm inFigure 2 can be explained as follows. First, unigram feature extraction is performed by using the *GenerateUnigram (Data)* function. The *Data* variable is a list of sentences from the corpus of scientific papers. The next process is calculating the *ProportionWord* value for each word in *Unigram*. ProportionWord value is the division between the *TFCompClass* function and the *compare* sentences number in the whole sentence. The *ProportionWord* feature is then obtained, which is a collection of words in the *ProportionWordList* variable with *ProportionWord* value of at least 10%.

```
Procedure ProportionWordFeature (output ListFeatures1: array[] of String )
{ Output, collection of words that have a proportion-word value to total Hits > = 10%  }
Dictionary
    Type ElmtPropWord : <Word: String, Prop: Real>
    N                 : Integer
    ProportionWordList : array[] of ElmtPropWord
    Term              : array[] of String
    nCS               : integer { number of Compare Sentences}
    Unigram           : array[] of String
    function TFCompClass(String TermA) → integer
    {return Term Frequency TermA in "compare" class,  defined}
    function GenerateUnigram  (ListSentences: Array[] of String) → array[] of String
    {return list of uniqe word,  defined}
Algorithm
     Unigram ← GenerateUnigram(Data) {Data: list of sentences, defined}
     N ← Unigram.size
     For i=1 to N
        ProportionWordList.Prop[i]← TFCompClass(Term[i]) / nCS)
     ProportionWordList.Prop[i].sort(desc)
     ListFeatures1 ← ProportionWordList.Word[:10%]
```

Figure 2. Extraction Algorithm of ProportionWord Feature

B. ProbabilityWord Feature

The Probability Word feature is a binary word feature and contains a collection of words with high word probability values in *compare* class. This probability value is calculated based on comparison between the word-A occurrences number in *compare* sentence to total word-A occurrences number in whole sentence. The probabilitWord value (word A) can be calculated using the formula 2.

$$\frac{\#\,word\ A\ in\ \ compare\ sentence}{\#\,word\ A\ in\ \ all\ \ sentence} \qquad (2)$$

The implementation of this feature extraction procedure produces a collection of word features as contained in Table 4. As in the *ProportionWord* feature extraction process, the lower limit of probability value in the *ProbabilityWord* feature extraction process is also 10% with the same consideration as the previous process. The use of this feature can generate words contained in the *ProportionWord* feature. Words contained in the previous feature will be removed. The collection of words obtained from this extraction process can be seen in in the Table 4.

Table 4. List of Words in ProbabilityWord Feature

Nb.	Word	Nb.	Word
1	information	7	state-of-the-art
2	words	8	improvements
3	differs	9	significantly
4	compared	10	similar
5	cbc	11	state-of-the-art
6	mellish		

Detailed algorithm to extract *ProbabilityWord* feature can bee seen in Figure 3. The *ProbabilityWord* feature extraction process is carried out in several stages. In the first stage, the *Unigram* feature is generated by calling the *GenerateUnigram* function. Then for each word in the *Unigram* feature, a *probabilityWord* calculation is performed which is the division between the values of *TFCompClass (Term [i])* and *TF_AllClass Term [i]*. Duplicate features of the *ProportionWord* feature are then removed. The *ProbabilityWord* feature is a collection of words that have a *probabilityWord* value with a minimum value of 10%.

```
Procedure ProbabilityWordFeature (output ListFeatures2: array[] of String)
{collection words that have a probability > = 10% }
Type ElmProb : <Word: String, Prob: Real>
Dictionary
    ProbabilityWord      : array[1..N] of ElmProb
    ListProbabilityWord : array[1..N] of ElmProb
    Unigram              : array[] of String
    Procedure ProportionWordFeature (output ListFeatures1: array[] of String )
    function TFCompClass(String TermA) → integer
    {return Term Frequency of TermA in compare class}
    function TF_AllClass(String TermA) → integer
    {return Term Frequency of TermA in all class}
    function RemoveDuplicate(LX: array[1..K] of String, X: String)
    {Remove  X in LX, return cleaned LX;  K: size of LX}
    function GenerateUnigram  (ListSentences: Array[] of String) → array[] of String
Algorithm
    Unigram ← GenerateUnigram(Data) {Data: list of sentences, defined}
    ProportionWordFeature (ListFeatures1)
    N ← Unigram.size
    For i=1 to N
        ProbabilityWord.Prob[i]= TFCompClass(Unigram[i])/ TF_AllClass(Unigram [i])
        RemoveDuplicate(ListFeatures1, ProbabilityWord[i].Word)
    ListProbabilityWord ←  ProbabilityWord.Prob.sort(desc)
    ListFeatures2 ← ListProbabilityWord.Word[:10%]
```

Figure 3. Extraction Algorithm of ProbabilityWord Feature

C. CuePhraseWord feature

This feature was automatically generated based on the learning process from the data in the form of list of sentences. It combines the concepts of the two previous features. i.e. uses the initial features based on the ProportionWord feature and uses the ProportionWord feature to form a cue phrase. The resulting cue phrase is a collection of words selected based on the ProbabilityWord value as used in determining the ProbabilityWord feature. In general, there are 2 calculation steps or processes to obtain this feature. The three steps can be seen in detail in the cuePhraseWord procedure in Figure 4.

The CuePhraseWord feature extraction process in Figure 4 starts by calling the ProportionWordFeature procedure, from which the ProportionWord feature will be obtained. For each word feature in ProportionWord, the corresponding cue phrase is then determined using the GetCuePhrase function. The final step is to save this collection of cue phrases using the GetCuePhrase function. This GetCuePhrase function is a defined function in the form of a disjunction of several words that has a minimum probability value of certain probability to total Hits. The Probability value between X (in ProportionWord feature) and Bi (in Unigram feature is calculated by using formula (3).

$$P(X, B_i) = \frac{\#(X \text{ and } B_i) \text{ in compare sentence}}{\#(X \text{ and } B_i) \text{ in all sentence}} \quad (3)$$

Type
cue phrase : disjunction of Strings
Procedure CuePhraseWord (output ListFeatures3: array[] of cue phrase
Dictionary
K : integer
ProportionWord : Array[] of String
Procedure ProportionWordFeature (output ListFeatures1: Array[] of String)
Function GetCuePhrase(X: String) → cue phrase
{ $GetCuePhrase(X) \leftarrow B_1 \, OR \, B_2 \, ... OR \, B_N$ }
{ $B_1, B_2, ..., B_N$: terms in Unigram AND $P(X, B_i) \geq 0.2$ }
{ $P(X, B_i)$: probability to total Hits between X and B_i }
Algorithm
ProportionWordFeature (ProportionWord)
K = ProportionWord.size
For i=1 to K
ListFeatures3[i]← GetCuePhrase (ProportionWord [X_i]

Figure 4. Extraction Algorithm of CuePhraseWord Feature

D. CitationAll Feature

This is a further development of the citation feature used in previous research. It is a binary feature that shows the existence of citation in a sentence which can appear explicitly through the existence of citation mark or implicitly. The development carried out on this feature is citation detection of the previous sentence.

5. Experimental Result

In term of the dataset, this study used the *compare relation* sentence dataset derived from 75 scientific papers as used by Khodra et al.[12] and Sibaroni et al. [13]. The data has been labeled with the rhetoric category according to the rhetoric category of (Teufel et al., 2009). There are 15 rhetoric categories where among those categories there are 2 rhetoric categories, namely CODI and ANTISUPP which have the potential to be *compare relation* category. The annotation process is then carried out on these 2 labels, because not all of these categories contain paper relations. There were 12760 sentences of which 95 were labelled "*compare*" while the rest were "*non-compare*". For the validation method, this research uses 10-cross-validation method as used in text classification research.

Three experimental scenarios were constructed, and they are:

1. Testing the baseline and intersection features between baseline features
2. Testing each new feature separately
3. Testing all the combined new features and baselines

Testing of Park and Blake's baseline features and 2 other scenarios was conducted using the supervised machine learning approach with 6 classifiers involving IBk (also known as K-Nearest Neighbours classifier), Logistic Regression, Naïve Bayes, Bayesian Network, Support Vector Machine (SVM), and Decision Tree (DT). In general, these have often been applied in text classification, e.g. the application of IBk by [9], [14], Logistic regression by [14]–[16], Decision Tree by [3], [17], [18], etc. Among these classifiers, SVM [19]–[21] and Naïve Bayes[22], [23] are the most frequently used in the text classification studies and have a relatively higher performance than others.

Naïve Bayes used in this study included 3 variations i.e. Naive Bayes, with Supervised Discretization, and with Kernel Estimator. Naive Bayes with the Kernel estimator uses the kernel Gaussian function for the calculation of Conditional probabilities: $P(Xi \mid Ci)$, which is a probability that the feature of the Xi gave class Ci [24], [25]. Meanwhile, Naive Bayes with Supervised Discretization has a data discretization process performed by considering the *compare* target class [26].

The use of many classifiers in this paper was aimed at showing that the proposed features have a positive influence on all of them. In addition, it also focused on new features that are important to the process of relation identification. However, LightSide tool made by Mayfield *et al.* [27] was used in the classification process.

A. Testing baseline and combined baseline features

This was aimed at obtaining the best results through the use of baseline features in classifying *compare relation* sentences. The features applied include Wang et al.'s, Park and Blake's, and the combination of both.

The result was very influential in the next scenarios. When the results were very good, there was no need to develop new features but when they were not, the development of new features was absolutely necessary, and it became important to perform the second scenario. The experiment results of the baseline features can be seen in more detail in Table 5. In this study, performance is measured using F-Measure rather than accuracy, this is because the focus of the classification results is the *compare* category. When using accuracy, the performance of non-*compare* classes is also calculated so that the results obtained are biased.

The experiment results in Table 5 indicate that the application of both Wang et al.'s cue phrase and Park's feature has not provided satisfactory results. In some classifiers, the use of the baseline features has not even been able to classify the *compare relation* sentences correctly. In others, Park's feature provides a better result than Wang et al.'s, while the combination of both provides better performance than the single performance of each. This shows that the application of these two features is mutually reinforcing. The results obtained from the baseline feature also show that the performance is still quite low, therefore, it is very necessary to develop new stronger features.

Table 5. F-Measure values of the Baseline Feature and its Combination

Classifier	Wang et.al. feature	Park &Blade feature	Combination Wang et.al, Park & Blade
IBK	0.00	0.04	0.07
LR	0.00	0.00	0.00
NB	0.14	0.14	0.15
BN	0.00	0.06	0.13
SVM	0.00	0.05	0.08
DT	0.00	0.00	0.00
Mean	0.02	0.05	0.07

B. Testing of every new feature

This was aimed at checking whether each proposed new feature has a positive effect or not. This was performed separately and simultaneously, and the new features tested include *ProportionWord*, *ProbabilityWord*, and *cuephraseWord* features. The results can be seen in Table 6.

Table 6. F-Measure values of the separate and simultaneous baseline features

Classifier	Feature			
	1	2	3	4
IBK	0.15	0.06	0.15	0.17
LR	0.13	0.10	0.11	0.22
NB	0.27	0.22	0.17	0.14
BN	0.24	0.08	0.25	0.20
SVM	0.24	0.08	0.22	0.28
DT	0.22	0.00	0.13	0.17
Mean	0.21	0.09	0.17	0.21

where

1:`
ProportionWord and *CitationAll* feature
2:
ProbabilityWord and CitationAll feature
3:
cuePhraseWord *and CitationAll* feature
4:
combination 1,2 and 3 features

Table 6 shows that feature 1 (*ProportionWord* and *CitationAll*) and feature 4 (the combination of 3 features) have the same average value. Feature 1 has a relatively high level of performance compared to others. All the feature groups showed high performance of each classifier but the combination of 3 features group generally dominates.

These results also show that the new features proposed in this study are proven to provide a performance improvement that is far better than those of the best baseline features as presented in Table 5 because average increase in its overall performance doubles those of the baseline features, while the best feature performance increases more than 85% compared to those of baseline features.

C. Testing the combination of all new features and baselines

This third scenario was performed to see whether the combination of new features and baseline features could provide better performance or not. It was expected that this would provide a better result compared to using the features separately. Another goal of combining these features was to gain the highest performance compared to the performance obtained from the use of new features only. The results are presented in Table 7.

Table 7. F-Measure values of features combination vs. new features

Classifier	New feature	Combination of new feature and Baseline
IBK	0.17	0.20
LR	0.22	0.24
NB	0.27	0.29
BN	0.25	0.27
SVM	0.28	0.30
DT	0.22	0.19
Mean	0.23	0.25

The experimental results as presented in Table 7 indicate that the combination of the baseline and proposed new features have a positive effect on the performance of almost all

classifiers. The average performance increase in each classifier is 2%. The results also suggest that the proposed group is essential for the process of identifying *compare relation* as compared to the prior ones. However, the existence of the baseline features cannot be ignored because when they are combined with the proposed new features, a positive effect is observed.

Based on observations of the values of precision and recall for both the baseline feature and the proposed feature, the increase in the F-Measure value obtained is greatly influenced by the increase in the precision value. The proposed new feature is proven to significantly increase the precision value compared to the baseline feature. Low precision values obtained by the baseline feature indicate that the classifier model tends to classify non-*compare* sentences as *compare* sentences. This means that the features used in the baseline have not been able to identify the *compare* sentence precisely, because the feature apparently also still appears in the non-*compare* sentence.

6. Conclusion

In this study, we have successfully developed new important features that are shown to be essential in identifying the *compare* relation. These new feature groups are *ProportionWord*, *ProbabilityWord*, and *cuephraseWord*, that are automatically developed based on the proposed learning approach. The majority experimental results show that the use of all features in the classification process of *compare* sentences relation, provide the highest performance for each classifier. Moreover, the experiment results indicate that they can individually produce relatively higher performance in their classification system than the average performance of the best baseline feature group. In addition, the combination of all new and baseline features produces the highest classification performance for both the average performance of all classifiers as well as for the best classifier performance. This fact shows that all feature groups, i.e. both from the baseline as well as the newly proposed features in this study, do not contradict each other, but instead they reinforce each other in the process of the classification of *compare* sentence relation. The experiment result also shows that the increase in the F-Measure value in the proposed features is most influenced by the precision value. The proposed new feature has a significant increase in the precision value compared to the baseline feature. This shows that classifier model produced by the baseline feature tends to classify *non-compare* sentences as *compare* sentences.

7. References

[1]. N. J. Van Eck and L. Waltman, "CitNetExplorer : A new software tool for analyzing and visualizing citation networks," *J. Informetr.*, vol. 8, no. 4, hal. 802–823, 2014.

[2]. N. Jindal, B. Liu, and L. Bing, "Mining comparative sentences and relations," *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '06*, vol. 21, no. 2, hal. 1331–1336, 2006.

[3]. S. K. Saritha, "Methods for Identifying Comparative Sentences," *Comput. Appl.*, vol. 108, no. 19, hal. 23–26, 2014.

[4]. W. Wang, P. Villavicencio, and T. Watanabe, "Analysis of reference relationships among research papers, based on citation context," *Int. J. Artif. Intell. Tool*, vol. 21, no. 2, hal. 1–24, 2012.

[5]. N. B. Prakash, D. Selvathi, and G. R. Hemalakshmi, "Development of Algorithm for Dual Stage Classification to Estimate Severity Level of Diabetic Retinopathy in Retinal Images using Soft Computing Techniques," *Int. J. Electr. Eng. Informatics* -, vol. 6, no. 4, hal. 717–739, 2014.

[6]. D. Park and C. Blake, "Identifying comparative claim sentences in full-text scientific articles," *Proc. 50th Annu. Meet. Assoc. Comput. Linguist.*, no. July, hal. 1–9, 2012.

[7]. D. H. Widyantoro and I. Amin, "Citation Sentence Identification and Classification for Related Work Summarization," *Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS), Jakarta, Indones.*, 2014.

[8]. S. Teufel and A. Athar, “Detection of Implicit Citations for Sentiment Detection,” *Proc. ACL-12 Work. Discov. Struct. Sch. Discourse, Jeju Island, South Korea, 2012*, no. July, hal. 18–26, 2012.
[9]. S. Teufel, A. Siddharthan, and D. Tidhar, “Automatic classification of citation function,” *Proc. EMNLP-06, Sydney, Aust.*, 2006.
[10]. Angrosh, S. Cranefield, and N. Stanger, “A Citation Centric Annotation Scheme for Scientific Articles,” *Proc. Australas. Lang. Technol. Assoc. Work. pages 5−14*, hal. 5–14, 2012.
[11]. M. De Marneffe *et al.*, “Universal Stanford Dependencies : A cross-linguistic typology,” *Proc. Ninth Int. Conf. Lang. Resour. Eval.*, 2014.
[12]. M. L. Khodra, D. H. Widyantoro, and E. A. Aziz, “Automatic Tailored Multi-Paper Multi Paper Summarization based on Rhetorical Document Profile and Summary Specification,” *J. ICT Res. Appl.*, vol. 6, no. 3, hal. 220–239, 2012.
[13]. Y. Sibaroni, D. H. Widyantoro, and M. L. Khodra, “Survey on research paper’s relations,” in *2015 International Conference on Information Technology Systems and Innovation, ICITSI 2015 - Proceedings*, 2016.
[14]. Y. Sibaroni, D. H. Widyantoro, and M. L. Khodra, “Extend Relation Identification in Scientific Papers Based On Supervised Machine Learning,” in *The 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS 2016)*, 2016.
[15]. D. H. Widyantoro, M. L. Khodra, B. Riyanto, and E. A. Aziz, “A Multiclass-based Classification Strategy for Rhetorical Sentence Categorization from Scientific Papers,” *J. Inf. Commun. Technol.*, vol. 7, no. 3, hal. 235–249, 2013.
[16]. D. ´O S´eaghdha and S. Teufel, “Unsupervised learning of rhetorical structure with un-topic models,” *Proc. COLING 2014, 25th Int. Conf. Comput. Linguist. Tech. Pap. pages 2–13, Dublin, Ireland, August 23-29 2014*, 2014.
[17]. N. Webb and M. Ferguson, “Automatic Extraction of Cue Phrases for Cross-Corpus Dialogue Act Classification,” no. August, hal. 1310–1317, 2010.
[18]. D. Kaplan, R. Iida, and Takenobu Tokunaga, “Automatic Extraction of Citation Contexts for Research Paper Summarization : A Coreference-chain based Approach,” *Work. Text Cit. Anal. Sch. Digit. Libr. ACL-IJCNLP 2009*, no. August, hal. 88–95, 2009.
[19]. J. Meng, H. Lin, and Y. Yu, “A two-stage feature selection method for text categorization,” *Comput. Math. with Appl.*, vol. 62, no. 7, hal. 2793–2800, 2011.
[20]. B. Aljaber, D. Martinez, N. Stokes, and J. Bailey, “Improving MeSH classification of biomedical articles using citation contexts,” *J. Biomed. Inform.*, vol. 44, no. 5, hal. 881–896, 2011.
[21]. Kuspriyanto, O. S. Santoso, D. H. Widyantoro, H. S. Sastramihardja, K. Muludi, and S. Maimunah, “Performance Evaluation of SVM-Based Information Extraction using τ Margin Values,” *Int. J. Electr. Eng. Informatics* -, vol. 2, no. 4, hal. 256–265, 2010.
[22]. N. Tandon and A. Jain, “Citation Context Sentiment Analysis for Structured Summarization of Research Papers,” *35th Ger. Conf. Artif. Intell.*, hal. 98–102, 2012.
[23]. D. Isa, “Text Document Pre-Processing Using the Bayes Formula for Classification Based on the Vector Space Model,” hal. 79–90, 2008.
[24]. Y. Murakami and K. Mizuguchi, “Applying the Naïve Bayes classifier with kernel density estimation to the prediction of protein – protein interaction sites,” vol. 26, no. 15, hal. 1841–1848, 2010.
[25]. A. Pérez, P. Larrañaga, and I. Inza, “Bayesian classifiers based on kernel density estimation : Flexible classifiers,” *Int. J. Approx. Reason.*, vol. 50, no. 2, hal. 341–362, 2009.
[26]. P. Das and S. Sharma, “An Entropy Based Effective Algorithm for Data Discretization,” vol. 4, no. 3, 2017.
[27]. E. Mayfield and C. P. Rose, *LightSIDE: Open Source Machine Learning for Text. Handbook of automated essay evaluation: Current applications and new directions.* 2013.

Yuliant Sibaroni received the bachelor's degree in Statistics from Gadjah Mada University (UGM) in 1999. He received master degree in Informatics from Institut Teknologi Bandung (ITB) in 2008. Currently, he is a Ph.D student at the School of Electrical Engineering and Informatics, Institut Teknologi Bandung. Lastly, he works as a lecturer and researcher at Telkom University. His research interests include Machine Learning, Natural Language Processing and Text Mining.

Dwi Hendratmo Widyantoro is a Professor in Machine Learning. He received the bachelor's degree in Informatics from Institut Teknologi Bandung (ITB) in 1991. He received master degree in computer science in 1999 and Ph.D in 2003 from Texas A&M University. Since 1994, he has been with the School of Electrical Engineering and Informatics (STEI), Institut Teknologi Bandung, where he is currently a vice dean of Academic at STEI, Institut Teknologi Bandung. His research interests include Machine Learning, Information Summarization, Information Extraction and Information Retrieval.

Masayu Leylia Khodra received the bachelor's, in Informatics from Institut Teknologi Bandung (ITB) in 1999. She also received master in 2006 and doctoral degree in Informatic in 2012 from Institut Teknologi Bandung (ITB) . Since 2008, he has been with the School of Electrical Engineering and Informatics (STEI), Institut Teknologi Bandung. Her research interests include text classsification, text summarization, expert system, machine learning and artificial intelligence.