Phrase-Level Sentiment Polarity Classification Using Rule- Based Typed Dependencies

4 authors, including:

Luke Tan
Nanyang Technological University
9 PUBLICATIONS 123 CITATIONS

Jin-Cheon Na
Nanyang Technological University
67 PUBLICATIONS 1,120 CITATIONS

Kuiyu Chang
LinkSure China Holding Private Limited
75 PUBLICATIONS 1,957 CITATIONS

Phrase-Level Sentiment Polarity Classification Using Rule-Based Typed Dependencies

Luke K.W. Tan, Jin-Cheon Na and Yin-Leng Theng
Wee Kim Wee School of Communication and Information
Nanyang Technological University
31 Nanyang Link, Singapore 637718
{w080078,tjcna,tyltheng}@ntu.edu.sg

Kuiyu Chang
School of Computer Engineering
Nanyang Technological University
Block N4 Nanyang Avenue
Singapore 639798
{askychang}@ntu.edu.sg

ABSTRACT
The advent of Web 2.0 has led to an increase in user-generated content on the Web. This has provided an extensive collection of free-style texts with opinion expressions that could influence the decisions and actions of their readers. By detecting the opinion expressed, we can identify the sentiments on the topics discussed and the influence exerted on the readers. In this paper, we introduce an automatic approach in deriving polarity pattern rules to detect sentiment polarity at the phrase level, which can be generalized across domains. Recent sentiment analysis research has focused on the functional relations of words using typed dependency parsing, providing a refined analysis on the grammar and semantics of textual data. Heuristics are typically used to determine the typed dependency polarity patterns, which may not comprehensively identify all possible rules. We study the use of class sequential rules (CSR) to automatically learn the typed dependency patterns, and bench mark the performance of CSR against a heuristic method. We then introduce a generalized polarity pattern rules which allow the CSR polarity pattern rules to be applied further across different domains. Preliminary results show this leads to further improvements in classification performance achieving over 80% F1 scores for most test cases.

Categories and Subject Descriptors
I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *text analysis*; H.2.8 [**Database Management**]: Database Application – *data mining.*

General Terms
Algorithms, Performance, Languages.

Keywords
Sentiment analysis, class sequential rules, typed dependency.

1. INTRODUCTION
The growth and popularity of opinion-rich websites such as blogs and Twitter have presented new opportunities and challenges for researchers to extract opinionated information and sentiments from these sites. These sites allow individuals or group of individuals to express their thoughts, voice their opinions, and share their experiences and ideas, which could influence their readers. By acquiring the ability to detect sentiments expressed in the content, we can identify the chain of influence flow as observed in studies by Agarwal et al. [2] and Tan et al. [20]. As an example, the negative sentiment expressed by the phrase "*extensive damage*" in the sentence "*The oil spill caused extensive damage to marine and wildlife habitat.*" would cause readers to have a negative opinion of the oil spill. In this study, we explore sentiment analysis by using an automatic approach to derive polarity pattern rules and using the rules to detect sentiment polarity at the phrase level.

Sentiment analysis has been employed in various applications such as opinion-based search engines [1, 5], where the ability to search for topics with extreme sentiments could help governmental institutions find terrorist organizations using the web as a communication tool, especially in a terrorist attack, automatic detection of sentiments of financial blogs [6, 15] for market evaluation, including during a financial crisis, and identification of negative sentiments on companies or products [4, 13] in a commercial crisis where a lapse in delivery of quality services and products have led to bad reputation. The ability to detect sentiments in online content could identify the sentiment flow, where further stimuli could be added to aid the flow of positive information or pre-emptive or preventive actions taken to minimize any negative impact.

Sentiment analysis had been studied at both the document and sentence level with the goal of assigning an overall sentiment polarity for the document or sentence [18, 22]. Other studies had proposed methods to predict sentiment polarity at the clause or phrase levels to provide a more refined analysis [21, 23, 24]. In our study, we focus on sentiment analysis at the phrase level, which we specifically define as the bigrams in typed dependencies. Figure 1 shows examples of the adjectival modifier (AMOD), adverbial modifier (ADVMOD), and direct object modifier (DOBJ) typed dependencies for the respective phrases.

AMOD(calamity:NN, great:JJ) for "great calamity"
ADVMOD(destroy:JJ, completely:RB) for "completely destroy"
DOBJ(killed:VB, innocents:NN) for "killed innocents"

Figure 1: Typed dependency bigrams (NN: noun, JJ: adjective, RB: adverb, and VB: verb)

The extracted patterns are then used in a linguistic approach to determine the sentiment polarity of unseen bigram phrases. Previous linguistic approaches in sentiment analysis had leveraged on semantic dependencies between words to predict

SWSM '11, July 28, 2011, Beijing, China.

sentiments. Wilson et al. [24] observed that word patterns along with its modifiers could determine the intensity of phrase sentiments. For example, in the phrase "*indiscriminate* *killing*", the adjectival modifier "*indiscriminate*" gives the phrase an overall negative sentiment. Hence, the discovery of word patterns within typed dependencies could reveal clues to identifying the phrase polarity.

The fundamental notion of typed dependency is based on the idea that the syntactic structure of a sentence consists of binary asymmetrical relations between the words [14]. Intuitively, by using typed dependencies, the syntactic structure of the sentence will be taken into consideration during the sentiment analysis. A natural language parser is a program that works out the grammatical structure of sentences. For example, the grouping of words as phrases, and the identification of words that are the subject or object of a verb. In this study, we use the Stanford parser[1] to generate typed dependencies from our sentence dataset. Figure 2 shows a typed dependency tree generated from the Stanford parser for the sentence "*The cruel dictator willfully suppressed the citizens.*" The corresponding typed dependency bigrams are listed in Figure 3. In this example, the parser generates the typed dependencies output showing the semantic relationship between the bigrams. Typed dependencies facilitate the analysis of semantic relationships between words based on both their grammatical relationships and overall sentence syntactical structure. This approach allows words positioned far apart to be analyzed without neglecting the semantic and syntactic significance that could impact sentiment prediction performance.

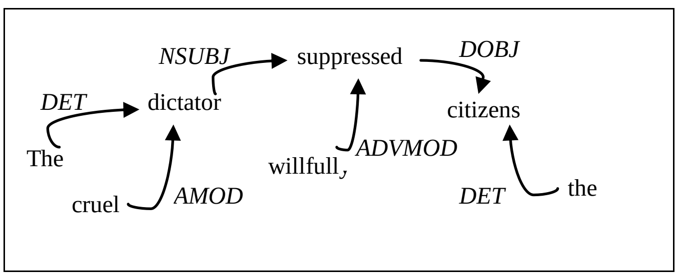

Figure 2. Typed dependency tree

DET(dictator-3, The-1)
AMOD(dictator-3, cruel-2)
NSUBJ(suppress-5, dictator-3)
ADVMOD(suppress-5, willfully-4)
DET(citizens-7, the-6)
DOBJ(suppress-5, citizens-7)

Figure 3. Generated typed dependencies

From Figure 3, we could see that by analyzing the polarity patterns of the bigram words, we can infer the polarity of the typed dependencies. For example, "AMOD(dictator-3, cruel-2)" could indicate a polarity pattern rule of "AMOD(NN(-),JJ(-))" giving a negative output, where (NN(-)) and (JJ(-)) refer to negative oriented noun term and adjective term respectively. In this paper, only the subjectivity of the AMOD, ADVMOD, and DOBJ typed dependencies are evaluated as they are deemed to contain the vast majority of opinionated expressions [8, 24]. In the Stanford parser, an adjectival modifier (AMOD) of a noun phrase is any adjectival phrase that serves to modify the meaning of the noun phrase, an adverbial modifier (ADVMOD) of a word is an adverb or adverbial phrase that serves to modify the meaning of the word, and the direct object (DOBJ) of a verb phrase is the object of the verb. The nominal subject (NSUBJ) typed dependency is defined as a noun phrase that is the syntactic subject of a clause. Though NSUBJ typed dependency can contain possible bigram polarity patterns leading to typed dependency polarity prediction, our focus in this study is on the subjectivity expressed on the object and not on the subject that expresses the sentiments. Subsequent studies would explore additional typed dependencies (i.e., other various syntactic structures) that could influence polarity prediction, and the use of typed dependencies to identify the subject with the subjectivity expressed.

A common approach to word pattern discovery is through heuristic, that is, knowledge engineering or manual discovery. Shaikh et al. [19] considered the semantic relationship between textual components in a sentence and the computation of contextual valence of the words to create word pattern rules. Thet et al. [21] manually created rules based on grammatical dependencies and prior sentiment scores of the feature terms to compute the sentiment of a clause. In contrast to a manual heuristic approach, our proposed approach could automatically generate a comprehensive set of rules to provide the sentiment classification. In our study, we extract the typed dependency bigrams of the sentences using the Stanford parser and apply Class Sequential Rules (CSR) proposed by Liu [11] to derive polarity class rules from sequential patterns within the bigrams. We then compare the CSR rules with the heuristic rules adapted from Thet et al. [21]. To the best of our knowledge, no previous studies have used CSR to automatically derive the sentiments of typed dependency bigrams.

The next section describes the related work followed by the research design where details of our model are given. Next, we present the evaluation process and results. This is followed by the discussion and conclusion.

2. RELATED WORK

Earlier studies in sentiment analysis at the sentence or phrase level had used the notion that an opinion word associated with its aspect or feature would appear in its vicinity [7, 10]. However, opinionated text could be written in elaborate styles where sentences have nested clauses with the related opinion and subject words hidden in separate clauses. Methods using word distances [7] or part-of-speech patterns [25] will not be able to detect the relationship of the opinionated words as these methods typically assume both the opinion and aspect words to appear within a certain distance of one another. Moreover, the grammatical relationship between the words had largely been ignored.

More recent sentiment analysis studies had used linguistic approach which leveraged on the semantic dependencies between words to predict sentiments. Wilson et al. [24] observed that specific word patterns alongside its modifiers could determine the intensity of private states including sentiments. In their study, bigrams (termed as bilex) that appears in a specific pattern involving a word and just one of its modifiers were included as features in the evaluation.

A common approach to extract patterns within word dependencies for sentiment detection was through heuristic means [12, 19, 21]. Moilanen and Pulman [12] proposed a sentiment composition model based on the concept that the global polarity of a sentence

[1] http://nlp.stanford.edu/software/lex-parser.shtml

is a function of the polarities of its parts. Specifically, the model combined two input constituents of any dependency type or size and calculated a global polarity for the resultant composite output constituent. For example, a rule ($OUT^{\alpha_{ij}} \rightarrow SPR^{\alpha_i} + SUB^{\alpha_j}$) includes the consequent (OUT), the superordinate (SPR), which is the stronger of the input constituents, and the dominated constituent subordinate (SUB). The polarity (α) of the OUT constituent was determined by the SPR constituent and the compositional processes executed by the SPR constituent on the SUB constituent. The model assumed non-neutral sentiment polarity over neutral polarity and handled negation through polarity reversal. Polarity resolution was achieved by ranking the input constituents based on their assigned weights.

Shaikh et al. [19] considered the semantic relationship between textual components in a sentence and the computation of contextual valence of the words to create word pattern rules. Semantic processing of the input text was based on the dependency analysis of each semantic verb frame, which is composed of the frame-invoking verb with its corresponding subject and object. Cognitive and common sense knowledge resources were used in the valence scoring of words. Rules were then used to calculate contextual valence to support word sense disambiguation, assess the valence of the semantic verb frame, and to assign overall valence to the whole sentence. Examples of rules are [ADJ_{pos}+(CON_{neg} | NE_{neg})→(negative output)] (e.g., "*strong cyclone*") and [ADJ_{pos}+(CON_{pos} | NE_{pos})→(positive output)] (e.g., "*brand new car*") where the adjectives (ADJ), concepts (CON), and named entities (NE) were assigned valence groupings.

Thet et al. [21] proposed the use of heuristic rules to compute the sentiment of a clause from the prior sentiment score assigned to individual words, taking into consideration the clause grammatical relations. Prior sentiment scores of the words were assigned using a domain specific and a generic opinion lexicon, while clauses were derived from dependency trees created from sentence parsing. The contextual sentiment scores of each clause was then inferred with heuristic rules by using the grammatical dependencies and prior sentiment scores of the sentiment and feature terms. An example rule is [(ADJ_{pos}+N_{pos})→(positive output)] (e.g., "beautiful art") where ADJ(adjective) and N(noun) are the respective feature term prior sentiment scores.

In general, heuristic approaches to creating polarity pattern rules may not provide substantial rule coverage. On the other hand, our proposed automatic approach can generate a very comprehensive set of rules to provide better sentiment classification performance. Further, the polarity pattern rules could be generalized across different domains. Later, we will show that our polarity feature performs on-par, if not better, compared to the heuristic or POS-polarity feature approach.

3. RESEARCH DESIGN

Our study combines both linguistic and machine learning methods to automatically detect the sentiment polarity of bigrams in specific typed dependencies. We parse the typed dependency bigrams of the sentences from the dataset and apply the CSR algorithm to derive polarity rules from sequential patterns within the bigrams. We then evaluate the CSR rules with the heuristic rules adapted from Thet et al. [21]. Figure 4 gives an overview of our CSR and HR approach.

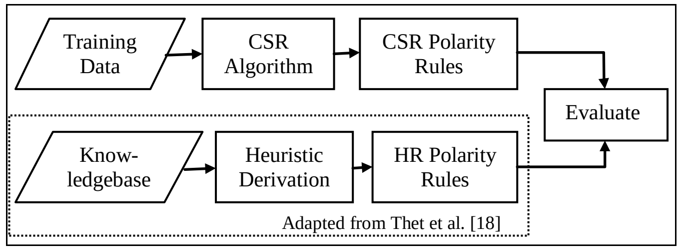

Figure 4: Overview of the CSR and HR approach.

3.1 Dataset

We use a readily available subjectivity dataset from Pang and Lee [16] to obtain 1000 records each for the adjectival modifier (AMOD), adverbial modifier (ADVMOD), and direct object modifier (DOBJ) typed dependencies.

We generate the typed dependencies from sentences in the dataset using the Stanford parser and derive the polarity pattern rules through the semi-supervised CSR algorithm. Next, we tag the polarity of each word in the typed dependencies bigrams by matching the words to the subjectivity lexicon terms from Thet et al. [21], which were derived from SentiWordNet [2] and the subjectivity lexicon of Wilson et al. [23]. Each word in the typed dependencies bigrams is tagged one of three polarity values of positive (+), negative (-), and neutral (0). Table 1 shows the distribution of the subjectivity lexicon terms with respect to their polarity and part-of-speech (POS), where it is observed that there are generally more negative terms than positive terms, except for the adverb terms.

Table 1. Polarity distribution of the subjectivity lexicon terms

POS	Positive (+)	Negative (-)	Neutral (0)
Adjectives	5375	8392	1027
Adverbs	2010	934	305
Verbs	1149	2303	714
Nouns	5407	8591	1409

The typed dependencies, along with the polarity tagged bigram words, are then manually annotated to give the overall typed dependency polarities. Manual annotation of the typed dependency bigrams' polarity between two annotators gives a Cohen Kappa [3] value of 0.78, indicating an acceptable reliability of the coded polarities. The respective typed dependency bigrams' polarity distributions in the coded dataset are shown in Table 2. It is observed for the dataset that more opinions are expressed using adjectival words as seen from AMOD having the most sentiment phrases, while DOBJ has the least number.

Table 2: Polarity distribution of bigrams in coded dataset.

Type	Positive (+) Class	Negative (-) Class	Neutral (0) Class	Total
AMOD	320	228	452	1000
ADVMOD	245	222	533	1000
DOBJ	196	153	641	1000

Figure 5 shows samples of the coded data records which indicate the relation "typed-dependency(governor-term:POS[polarity], dependent-term:POS[polarity])→(bigram-polarity)". Acronyms

[2] http://sentiwordnet.isti.cnr.it

used here are POS:part-of-speech, NN:noun, JJ:adjective, RB:adverb, and VB:verb.

```
AMOD(movie:NN, nice:JJ+)→(+)
ADVMOD(nice:JJ+, very:RB+)→(+)
DOBJ(upset:VB-, viewers:NN+)→(-)
```

Figure 5: Sample generated output of coded data.

Wilson et al. [24] found that the sparsity of word occurrences in clauses posed a problem in extracting meaningful patterns. Joshi and Penstein-Rosé [9] observed that generalizing the typed dependency words into their part-of-speech improves performance. We further generalized the typed dependency bigram features to their respective part-of-speech and polarity (e.g., JJ+). This reduces the number of distinct features, which increases the statistical significance of the patterns. We further evaluated the effects of generalization by reducing the features to just the polarity of each word in the bigram. Our results show that the performance of using only the polarity of terms is on-par to that of using both part-of-speech and bigram polarity. It is noted that the use of sentiment scores in our evaluation would have provided pattern rules at sentiment score level. This could be explored in future work with a large dataset. Figure 6 shows samples of the generated output of the POS-polarity records, in the form of "typed-dependency(governor-POS[polarity], dependent-POS[polarity])→(bigram-polarity)".

```
AMOD(NN(0),JJ(+))→(+)
ADVMOD(JJ(+), RB(+))→(+)
DOBJ(VB(-), NN(+))→(-)
```

Figure 6: Sample generated output for POS-polarity evaluation.

Figure 7 shows samples of the generated output for the polarity records, indicating the relation "typed-dependency (governor[polarity],dependent[polarity])→(bigram-polarity)".

```
AMOD((0),(+))→(+)
ADVMOD((+), (+))→(+)
DOBJ((-), (+))→(-)
```

Figure 7: Sample generated output for polarity evaluation.

At the phrase level, i.e., bigram words, and given a comprehensive subjectivity lexicon, the generated polarity pattern rules can be applied regardless of domain. For example, AMOD(movie:NN, bad:JJ-)→(-) or AMOD(dictator:NN, bad:JJ-)→(-) from a movie review and news blog respectively, will give similar negative polarity output in the pattern rule evaluation. In addition, by ignoring the bigram, which can be domain dependent, and the part-of-speech, and considering only its polarity as a feature in the polarity evaluation, we further generalize our model across domains.

3.2 Class Sequential Rules (CSR)

We adapt Liu's [11] class sequential rules (CSR) to identify the pattern rules for each bigram's polarity class with our implementation given as follows.

- Let S be the set of relation data sequences where each sequence is also labelled with a class y. Let I be the set of all items in S, and Y={(+), (-), (0)} be the set of all class labels, and $I \cap Y = \varnothing$.
- The input data D is denoted as $\{(s_1, y_1), (s_2, y_2), \ldots, (s_n, y_n)\}$, where s_i is a sequence in S and $y_i \in Y$ is its class. The sequence s_i is represented as a relational triplet for the item records: [item1(POS, polarity), item2(POS, polarity), item3(class)] and [item1(polarity), item2(polarity), item3(class)] for the two respective evaluations. POS denotes the part-of-speech tag.
- A **class sequential rule** is of the form $X \rightarrow y$, where X is a sequence, and $y \in Y$. A data instance (s_i, y_i) is said to **cover** a CSR, $X \rightarrow y$, if X is a subsequence of s_i. A data instance (s_i, y_i) is said to **satisfy** a CSR if X is a subsequence of s_i and $y_i = y$.

From the CSR algorithm, we extract both unigram and bigram pattern rules based on an experimentally determined minimum support threshold of 0.01 as suggested by Hu and Liu [7]. Additionally, we use a minimum confidence threshold of 0.5 to remove noisy data. The generated rules are sorted in order of descending F1 measure of their support and confidence values where:

$$F1 = \frac{2 * Support * Confidence}{Support + Confidence}$$

3.3 Heuristic Rules

We adapted the heuristic rules implementation by Thet et al. [21] as our baseline. The heuristic rules (HR) used in our study are listed in Table 3. We further separate the neutral class rules, which were assumed positive in Thet et al [21] to make a direct comparison with the CSR's three classes.

Negation was considered in the heuristic rules in Thet et al. [21] to improve the sentiment scoring for the bigrams. This was done by triggering the negation rules when the bigrams match a list of pre-identified negating terms. For example, the negating adverbs such as 'hardly' or 'rarely' could change the original sentiment orientation of a verb or an adjective. In the phrase 'hardly fail', the negation of the negative verb 'fail' by the negating term 'hardly' would make the phrase positive. On the other hand, the phrase "rarely pass" is not as positive as "hardly fail" and could be classified as negative. We observed that there exist more complex relationships between words that could influence the polarity of the typed dependency bigrams. This includes negation and other relationships, which are discussed later.

Table 3. Heuristic pattern rules (HR)

HR Type	(+) class	(-) class	(0) class
AMOD (governor, dependent)	NN+, JJ+ NN, JJ+ NN+, JJ	NN-, JJ+ NN-, JJ NN+, JJ- NN, JJ- NN-, JJ-	NN, JJ
ADVMOD (governor, dependent)	VB+, RB+ VB, RB+ VB+, RB	VB-, RB+ VB-, RB VB+, RB- VB, RB- VB-, RB-	VB, RB
DOBJ (governor, dependent)	VB+, Obj+ VB+, Obj VB, Obj+	VB+, Obj- VB, Obj- VB-, Obj+ VB-, Obj VB-, Obj-	VB, Obj

4. EVALUATIONS

Ten-fold cross validation for each of the positive (+), negative (-), and neutral (0) class was conducted for the CSR method. The apportioned training data is used to generate the polarity pattern rules and in turn tested on the other portion of test records. The heuristic rules from Thet et al. [21] are coded into the evaluation method with each record processed by the heuristic rules. The performance of the CSR rules are then compared with the modified heuristic method adapted from Thet et al. [21]. We compared in the following sections each of the three major typed dependencies for CSR and HR.

4.1 AMOD Evaluation

The list of generated AMOD CSR rules are shown in Table 4, while the top 10 generated AMOD CSR rules with respect to their F1 scores are listed in Table 5. The rules are in the order of "governor-POS(polarity),dependent-POS(polarity)", with the unigram pattern rules represented as "governor-POS,_" and "_,dependent-POS".

Table 4. CSR rules for AMOD typed dependency: AMOD(governor, dependent)

(+) class patterns	(-) class patterns	(0) class patterns
• NN+,_ (beauty) • JJ+,_ (interesting)	•NN-,_ (problem)	• VB,_ (known) • CC,_ (or)
• _, JJ+ (great) • _,VB+ (winning)	• _,JJ- (forgettable) • _,VB- (deteriorating)	• _,JJ (recent) • _,VB describe) • _,NN (story) • _,DT (the)
• NN, JJ+ (movies, greatest) • NN+,JJ+ (insight, great) • NN+, JJ (accomplished, most) • NN,VB+ (vision, expanding)	• NN, JJ- (concept, stale) • NN-, JJ- (struggle, desperate) • NN-, JJ (poignancy, certain) • NN-, JJ+ (problem, biggest) • NN+, JJ- (tribute, hollow) • NN, VB- (flaws, infuriating)	• NN, JJ (years, recent) • NN, VB (member, surviving) • NN, DT (ending, the)

Table 5: Top 10 AMOD CSR rules.

#	F1	Governor	Dependent	Class
1	1.00	NN (member)	VB(surviving)	0
2	1.00	VB (known)	_	0
3	0.99	NN-(poignancy)	JJ- (certain)	-
4	0.99	NN(vision)	VB+(expanding)	+
5	0.99	NN(subplot)	VB-(baffling)	-
6	0.98	_	VB+(rising)	+
7	0.95	NN+ (insight)	JJ+ (great)	+
8	0.85	NN (concept)	JJ- (stale)	-
9	0.81	NN- (problem)	_	-
10	0.80	NN (movie)	JJ+ (great)	+

In comparison with HR patterns of Table 3, CSR discovered additional patterns containing verbs like "*expand*" (VB+), "*baffling*" (VB-), and "*known*" (VB). These are in fact adjectival verbs, which explain their presence in AMOD. The addition discovered pattern rules show CSR's capability to provide greater in-depth analysis of typed dependency bigram patterns. However, as with any rule-based approach, the ordering of rules must be deliberated carefully. Simply ordering them based on F1 values poses a problem in CSR because a generic unigram pattern rule of higher priority may inadvertently override a more specific bigram rule of lower priority from a different class. To overcome this, we allow the specific bigram pattern rules to take precedence over generic unigram rules. For example, the more specific bigram pattern rule AMOD(NN+,JJ-)→(-) with F1=0.53 overrides the unigram pattern rule AMOD(NN+,_)→(+) with a higher F1=0.61 in our implementation.

Table 6. Results for AMOD POS-polarity and polarity rules

Method	Class	Precision (%)	Recall (%)	F1 (%)	AvgF1 (%)
CSR (POS-polarity)	(+)	78.20	96.84	86.37	84.97
	(-)	80.44	94.25	86.52	
	(0)	95.83	71.97	82.04	
HR (POS-polarity)	(+)	79.38	92.78	85.45	83.95
	(-)	81.77	91.80	86.09	
	(0)	89.18	73.34	80.31	
CSR (polarity)	(+)	78.56	98.92	87.42	85.37
	(-)	79.31	96.52	86.75	
	(0)	98.66	70.46	81.95	
HR (polarity)	(+)	79.41	98.92	87.95	85.87
	(-)	79.58	98.31	87.72	
	(0)	98.66	70.45	81.95	

As seen in Table 6, the F1 values for the 4 methods, POS-polarity and polarity features for both CSR and HR approach are not significantly different. However, CSR found additional rules such as AMOD(NN,VB+) →(+), AMOD(NN,VB-)→(-) and others for the neutral (0) class, which are absent from HR. As a result, CSR enjoys up to 4% better recall (96.84 versus 92.78) for the polarized classes, and 6% better precision for the neutral (0) class (95.83 versus 89.18), compared to HR.

From Table 6, although the increase in average F1 for the polarity features method is only 0.4% (85.37 versus 84.97) for CSR and close to 2% (85.87 versus 83.95) for HR, it shows that part-of-speech is not essential in the bigram pattern rules. In fact, part-of-speech tagging introduces errors, lowering the recall (up to 6% lower for (+) and (-) classes) in HR.

Table 7: AMOD polarity rules

Sequence type	CSR patterns			HR patterns		
	(+) class	(-) class	(0) class	(+) class	(-) class	(0) class
Governor only	+, _	-, _	0, _	N.A.	N.A.	N.A.
Dependent only	_, +	_, -	_, 0	N.A.	N.A.	N.A.
Governor, Dependent	+, + 0, + +,0	-, - 0, - -, 0 +, -	0, 0	0, + +, + +, 0	0, - -, 0 -, - +, - -, +	0, 0

In Table 7, the HR AMOD(-,+)→(-) pattern is covered in CSR by the AMOD(-,_)→(-) unigram rule. However, the CSR AMOD(_,+)→(+) unigram rule with higher (F1=0.85) priority overrides the lower (F1=0.36) priority CSR AMOD(-,_)→(-) rule, which explains the lower positive (+) class precision and lower negative (-) class recall for CSR polarity method, when compared to HR as shown in Table 6. Therefore, the priority of the unigram pattern rules for different classes affects the performance when overlapping rules of lower priority are overridden. For example, the bigram AMOD(disaster[-], biggest[+]) should be classified as negative (-) class. However, the higher priority CSR AMOD(_,+)→(+) rule overrides the lower priority CSR AMOD(-,_)→(-) rule and classified the bigram as positive (+) based on the polarity of the dependent term "biggest".

4.2 ADVMOD Evaluation

The generated ADVMOD CSR pattern rules are listed in Table 8, with the top 10 generated ADVMOD CSR pattern rules with respect to their F1 scores given in Table 9. CSR generates more pattern rules than HR due to the greater variety of part-of-speech types found in ADVMOD. In the Stanford parser, the adverbial modifier could contain an adverb or an adverbial phrase, which contain words that operate adverbially and are not restricted to just adverbs. This leads to more possible pattern combinations in describing sentiment phrase for adverbial modifiers. The rules generated showed that CSR is able to detect significant pattern rules like ADVMOD(JJ-, RB-), which contain the different types of part-of-speech not discovered in HR.

Table 8. CSR generated pattern rules for ADVMOD typed dependency: ADVMOD(governor, dependent)

Positive class patterns	Negative class patterns	Neutral class patterns
• JJ+,_ (thoughtful) • VB+,_ (magnified) • NN+,_ (genius)	• JJ-,_ (unoriginal) • VB-,_ (torturing) • NN-,_ (lie)	• VB,_ (showed) • NN,_ (product) • JJ,_ (observed) • RB+,_ (much) • RB,_ (soon) • DT,_ (a) • IN,_ (on)
• _,JJ+ (important)	• _,RB- (unfortunately) • _,JJ- (crap)	• _,RB (only) • _,IN (at) • _,NN (movie) • _,VB (turn) • _,JJ (next) • _,DT (the)
• JJ+, RB+ (beautiful, incredibly) • JJ+, RB (serene, seemingly) • VB+, RB+ (delighted, undoubtedly) • VB+, RB (engrossing, so) • NN+, RB+ (fun, simply)	• JJ-, RB+ (intrusive, simply) • JJ-, RB (flashy, visually) • JJ-, RB- (dull, unspeakably) • VB-, RB+ (frustrates, constantly) • JJ, RB- (dull, lethally) • VB, RB- (dubbed, poorly) • JJ+, RB- (interesting, barely) • VB-, RB (forgotten, long) • NN-, RB (violence, only) • NN-, RB+ (labor, just) • JJ-, DT (bad, that) • VB-, RB- (mugs,mercilessly)	• VB, RB (find, only) • VB, RB+ (cut, just) • NN, RB+ (comparison, much) • NN, RB (boys, only) • JJ, RB (predictable, finally) • JJ, RB+ (similar, so) • VB, IN (finishing, at) • RB+, RB (much, as) • RB, RB (about, only) • JJ+, RB (aware, partly)

Table 9. Top 10 ADVMOD CSR rules

#	F1	Governor	Dependent	Class
1	1.00	NN(boys)	RB(only)	0
2	1.00	VB-(falling)	RB-(short)	-
3	0.99	VB(find)	RB(only)	0
4	0.93	JJ+(good)	RB+(really)	+
5	0.93	JJ-(stupid)	RB-(insanely)	-
6	0.93	JJ(untold)	RB(largely)	0
7	0.89	VB+(give)	RB+(actually)	+
8	0.84	JJ(paced)	RB-(poorly)	-
9	0.83	JJ+(aware)	RB(partly)	0
10	0.80	NN-(lie)	_	-

From Table 10, the ADVMOD heuristic rules' performance is significantly lower than the CSR performance. This is due to the variability of the part-of-speech (e.g., JJ, NN, and etc.) type records, which are not discovered in HR. Our implementation classifies all unmatched bigrams as neutral (0) class, which explains the high recall and low precision (54.49%) values for neutral (0) class in the HR POS-polarity method. Thet et al. [21], in their implementation, classified all unmatched bigrams using a generalized default rule that leveraged on the sentiment scores, which could have provided a better performance. HR polarity method performance improved significantly over the HR POS-polarity method, demonstrating the effectiveness of using the more general polarity features.

Table 10. Results for ADVMOD POS-polarity and polarity rules

Method	Class	Precision (%)	Recall (%)	F1 (%)	AvgF1 (%)
CSR (POS, polarity)	(+)	78.49	75.39	76.21	82.81
	(-)	82.36	94.72	87.81	
	(0)	86.62	82.58	84.42	
HR (POS, polarity)	(+)	40.00	23.42	29.20	41.57
	(-)	81.51	19.10	29.99	
	(0)	54.49	82.99	65.54	
CSR (polarity)	(+)	75.06	80.94	77.47	83.10
	(-)	80.75	98.15	88.38	
	(0)	89.58	78.34	83.45	
HR (polarity)	(+)	52.53	97.51	67.93	74.34
	(-)	80.75	98.15	88.38	
	(0)	98.98	50.70	66.73	

Table 11: ADVMOD polarity rules

Sequence type	CSR patterns			HR patterns		
	(+) class	(-) class	(0) class	(+) class	(-) class	(0) class
Governor only	+, _	-, _	0, _	N.A.	N.A.	N.A.
Dependent only	_, +	_, -	_, 0	N.A.	N.A.	N.A.
Governor, Dependent	+, + +, 0	-, - -, + -, 0 +, - 0, -	0, 0 0, +	+, + +, 0 0, +	-, - -, + -, 0 +, - 0, -	0, 0

The generated ADVMOD polarity rules are shown in Table 11. In the ADVMOD(0,+) pattern, which is classified neutral (0) for CSR and positive (+) for HR, though HR correctly identified positive (+) class bigrams such as ADVMOD (orchestrates, beautifully), the vast majority of bigrams matching this particular pattern are in fact the neutral (0) class, which only CSR was able to find. Examples of such neutral (0) class bigrams include ADVMOD(acted , mostly) and ADVMOD (built, entirely). Further analysis of the bigram word semantics is required to resolve polarity conflicts such as ADVMOD (0,+)→((+)|(0)), while a grid search on the best support and confidence thresholds could also lead to better rules.

4.3 DOBJ Evaluation

The CSR generated pattern rules are listed in Table 12, while the top 10 DOBJ CSR pattern rules are given in Table 13.

Table 12. CSR generated pattern rules for DOBJ typed dependency: DOBJ(governor, dependent)

Positive class patterns	Negative class patterns	Neutral class patterns
• VB+,_ (deliver)	• VB-,_ (hate)	• VB,_ (describe) • DT,_ (the) • NN,_ (movie)
• _,NN+ (insight) • _,JJ+ (alive) • RB+ (enough) • _,VB+ (love)	• _,NN- (headache) • _,JJ- (weird)	• _,NN (thing) • _,DT (a) • _,VB (makes) • _,CD (two) • _,JJ (other) • _,IN (at)
• VB, NN+ (give, blessing) • VB+, NN (good, effort) • VB+, NN+ (deserves, dignity) • VB, RB+ (make, much) • VB+, RB+ (enjoying, much) • VB, JJ+ (made, richer) • VB+, JJ+ (growing, more) • VB, VB+ (reading, love)	• VB, NN- (gives, unease) • VB-, NN (upset, viewers) • VB-, NN+ (lost, ability) • VB-, NN- (suffering, failure) • VB-, PRP (depress, you) • VB, JJ- (say, least) • VB-, DT+ (unrewarding, all) • JJ-, NN (clueless, combination) • VB-, DT (fooled, some) • JJ-, VB (surreal, dabbling)	• VB, NN (describe, vision) • VB, PRP (call, me) • VB, DT (catch, some) • VB, WP (does, what) • VB, CD (do, what)

Table 13. Top 10 DOBJ CSR rules

#	F1	Governor	Dependent	Class
1	1.00	VB(reading)	VB+(love)	+
2	1.00	VB+(growing)	JJ+(more)	+
3	1.00	VB(catch)	DT(some)	0
4	1.00	VB(describe)	NN(vision)	0
5	1.00	VB+(enjoying)	RB+(much)	+
6	1.00	_	JJ-(weird)	-
7	0.99	VB(act)	JJ(other)	0
8	0.99	VB(say)	JJ-(least)	-
9	0.93	VB(gives)	NN-(unease)	-
10	0.89	VB-(suffering)	NN-(failure)	-

The results for CSR POS-polarity and HR POS-polarity method are similar as seen from Table 14. The heuristic rules method generalized the part-of-speech of the dependents as part of a predicate. Hence, though CSR generated more rules with respect to the varied part-of-speech types detected, performance of CSR (AvgF1 = 80.95%) and heuristic rules method (AvgF1 = 79.54%) are not significantly different.

Table 14: Results for DOBJ POS-polarity and polarity rules

Method	Class	Precision (%)	Recall (%)	F1 (%)	AvgF1 (%)
CSR (POS-polarity)	(+)	76.61	54.73	62.84	80.95
	(-)	86.79	97.11	91.53	
	(0)	86.11	91.12	88.49	
HR (POS-polarity)	(+)	68.81	55.41	61.26	79.54
	(-)	85.80	95.45	90.05	
	(0)	85.96	88.80	87.32	
CSR (polarity)	(+)	75.18	56.20	63.72	81.45
	(-)	85.56	100	92.09	
	(0)	86.87	90.34	88.54	
HR (polarity)	(+)	70.24	99.56	82.22	88.09
	(-)	85.42	98.85	91.50	
	(0)	99.62	83.08	90.57	

From Table 14, we see significant improvement in performance by using polarity features (F1 of positive HR polarity rule = 82.22%) versus part-of-speech polarity features (F1 of positive HR POS-polarity rule = 61.26%). Improvements in F1 for HR were across the board at around 8%, whilst improvements for CSR were marginal at around 0.5%. This could be due to the presence of unigram rules in CSR that overrides the lower and correct priority rules, which affect performance. While HR polarity contains a more comprehensive list of bigram pattern rules that give a high recall performance.

The DOBJ polarity rules are similar except for the CSR unigram rules and the HR(+,0)→(+) rule, which was not discovered in CSR method as shown in Table 15. The confidence of (_,0)→(+) is at 0.13, which is less than the minimum threshold of 0.5, and was not significant enough for the CSR(+,0)→(+) rule to be generated, and the pattern for (+,0)→(+) was covered using the unigram CSR(+,_)→(+) rule. However, using the more generic unigram rule caused a lower recall value for the CSR polarity positive (+) class compared with HR polarity positive (+) class because the neutral (0) class unigram rule CSR(_,0) →(0) has a higher F1=0.79 value, which overrides the positive unigram rule CSR(+,_)→(+) of F1=0.68 in classifying (+,0) patterns. This shows the dependency of CSR on the dataset polarity frequency.

Table 15. DOBJ polarity rules

Sequence type	CSR patterns			HR patterns		
	(+) class	(-) class	(0) class	(+) class	(-) class	(0) class
Governor only	**+, _**	**-, _**	0, _	N.A.	N.A.	N.A.
Dependent only	**_, +**	_, -	_, 0	N.A.	N.A.	N.A.
Governor, Dependent	+, + 0, +	0, - -, - +, -	0, 0	+, + **+, 0** 0, +	-, - **-, +** **-, 0** +, - 0, -	0, 0

5. DISCUSSION

In this study, we explore the use of the CSR method to derive the polarity predicting pattern rules and compared the pros and cons between the CSR and heuristic rules method. Our results show that CSR is capable of automatically generating a comprehensive list of bigram part-of-speech patterns compared to a heuristic approach, which improves subsequent sentiment classification performance. It was also shown that rules using polarity features perform better than those using part-of-speech polarity features, especially for the heuristic approach. In general, the heuristic rules perform similarly to the CSR rules, which could be due to the careful construction of the knowledge base. The exception is for ADVMOD typed dependency where the heuristic rules did not include rules for the varied types of part-of-speech due to a restricted scope to verb phrases. This shows that the heuristic approach is not able to detect new pattern rules beyond the knowledge base, while CSR is able to discover new and existing pattern rules based on evidential support of the pattern item sets using computational analysis. Further to that, the rule discovery process is automatic as no prior knowledge base is needed to generate the CSR rules. The discovery of new additional pattern rules generally improves the overall performance of the CSR model. Furthermore, they could be used in linguistics studies to further enhance the knowledge base on the semantic dependencies of bigrams. CSR is also able to generate unigram pattern rules that are more generic in scope as compared to the bigram pattern rules. The unigram rules are important as they could cover the less frequent patterns and improve the robustness of the CSR model by reducing the effect of noisy data. It is observed that unigram rules improve recall performance while bigram rules tend to improve precision performance.

We note that for CSR, the execution priority has to be given to the specific bigram rules over the generic unigram rules to maintain good performance. Furthermore, in CSR, the execution priority of the overlapping rules from different classes poses a problem. A generic but higher priority rule could override a specific and lower priority rule from another class, causing the test item to be wrongly classified by the more generic rule. Nonetheless, with all learning algorithms, the discovery of rules in CSR is dependent on the dataset used where the distribution and frequency of the polarity cases could determine the rules to be derived, and as a result impact the performance.

From the results, it is equally efficient to use the polarity sign as a feature term in the rules. This could be because there is a common polarity pattern for the various part-of-speech terms within the specific typed dependencies. The generalization of the feature terms leading to the corresponding increase in their frequency count has enhanced the significance of the important pattern rules, which improves the overall prediction performance. Moreover, evaluating just the polarity of words could increase the prediction model performance as additional part-of-speech tagging may introduce errors. Possible causes of error in the tests are the wrongly parsed typed dependencies such as AMOD(regard:VB, be:VB), and sentiment lexicon with wrong polarity errors in the typed dependency parsing holds at less than 5% and are controlled using the support and confidence threshold values, while the detected words with wrong polarity tags are corrected by the coders during the annotation process.

The generated polarity pattern rules can be generalized across domains as context has lesser effect at the bigram phrase level. Moreover, if we use just the polarity pattern rules, the bigram words, which can be domain dependent, are not considered during the polarity outcome evaluation. For example, a AMOD((-),(-)) polarity pattern rule will give the same negative output regardless of the domain analyzed. It is noted that, polarity conflict cases were systematically processed according to the priority of the generated rules in our study. A separate semantic analysis on the bigrams' words and part-of-speech to resolve polarity conflicts through identifying semantic relationships of the bigram words would have provided better performance.

Though our results showed that rules using generalized features performed well, analysis of the bigrams' semantics including the words and part-of-speech is still required to identify the more complex relationships between words, which are observed during our study that can influence phrase polarity. For example, negating terms (e.g., "*one cannot*"), domain terms (e.g., "*star trek*"), subjective terms appearing as neutral (e.g. "fiction" in "*science fiction*"), and neutral terms appearing as subjective (e.g., "red" in "*red carpet*"). This issue has been partially addressed by Moilanen and Pulman [12], who propose resolving the polarity conflict by weighing the importance of the constituents (bigram words) with respect to sentiment. The idea is to use the semantic importance of a word in the mixed sentiment phrase (e.g., "*impressively bad*") to resolve the polarity conflict. Other complex relationships between words include intensification and mitigation. Terms such as "*very*" intensifies the sentiments of its adjacent term (e.g. "*very good*"), while mitigators such as "*few*" reduces the sentiment (e.g., "*few support*"). Quirk et al. [17] described the effects of intensifiers as scaling upwards from an assumed norm, and identified two intensifier types: maximizers (e.g., absolutely, completely, and perfectly) and boosters (e.g., very much, a lot, and deeply). Mitigators are described as generally having a lowering effect on the force of the term and could be categorized into four groups: approximators (e.g., almost, nearly, and as good as), compromisers (e.g., kind of, sort of, quite, and rather), diminishers (e.g., mildly, partly, and somewhat), and minimizers (e.g., barely, hardly, and little). Our future work will focus on identifying these complex relationships to solve polarity conflicts in the phrase level to improve sentiment prediction performance. By improving the performance of sentiment analysis, we hope to extract more accurate opinions that are expressed in online content and detect the influence exerted on the readers.

6. CONCLUSION

Our study focused on sentiment analysis and evaluated the generation of typed dependency rules automatically and discussed the use of typed dependency rules for predicting phrase-level polarity. Though, performance for the CSR generated polarity pattern rules is good, error analysis revealed that there were polarity conflict situations which the polarity patterns rules cannot resolve. These polarity conflicts arise due to the possible lexical relationships that exist between words, which we would study in our future work. The key contributions of this paper are as follows. First, we propose using CSR to automatically derive the sentiments of typed dependency bigrams, which have never been tried before, to the best of our knowledge. Second, we study in detail the effectiveness of CSR derived rules for three major typed dependencies. Third, we systematically benchmarked our CSR approach with a Heuristic based approach. Finally, we discovered that using polarity rules provide results that are on-par, if not better compared to POS-polarity rules. This will improve the CSR and heuristic based approaches and remove the need to identify the POS. The results from this study would be useful for improving rule based approaches to sentiment analysis.

7. REFERENCES

[1] Abbasi, A., Chen, H., and Salem, A. 2008. Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. *ACM Trans. Inf. Syst.* 26, 3, Article 12 (June 2008), 1-34.

[2] Agarwal, N., Liu, H., Tang, L., and Yu, P. S. 2008. Identifying the influential bloggers in a community. In *Proceedings of the international conference on Web search and web data mining* (WSDM '08). ACM, New York, NY, USA, 207-218.

[3] Cohen, J. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20, 1, 37-46.

[4] Ding, X., Liu, B., and Yu, P.S. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the international conference on Web search and web data mining*(WSDM '08). ACM, New York, NY, USA, 231-240.

[5] Demartini, G. and Siersdorfer, S. 2010. Dear search engine: what's your opinion about...?: sentiment analysis for semantic enrichment of web search results. In *Proceedings of the 3rd International Semantic Search Workshop* (SEMSEARCH '10). ACM, New York, NY, USA, 1-7.

[6] Devitt, A. and Ahmad, K. 2007. Sentiment Polarity Identification in Financial News: A Cohesion-based Approach. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (Prague, Czech Republic, 2007). Association for Computational Linguistics, 984-991.

[7] Hu, M. and Liu, B. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (Seattle, WA, USA, 2004). ACM, 168-177.

[8] Jakob, N., Weber, S. H., Muller, M.C., and Gurevych, I. 2009. Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations. In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion* (Hong Kong, China, 2009). ACM, 57-64.

[9] Joshi, M. and Penstein-Rose, C. 2009. Generalizing dependency features for opinion mining. In *Proceedings of the ACL-IJCNLP*. Association for Computational Linguistics, 313-316.

[10] Kim, S.-M. and Hovy, E. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics* (Geneva, Switzerland, 2004). Association for Computational Linguistics, 1367-1373.

[11] Liu, B. 2006. *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*. Springer Berlin Heidelberg, New York, 37-54.

[12] Moilanen, K. and Pulman, S. Sentiment Composition. 2007. In *Proceedings of the Recent Advances in Natural Language Processing* (Borovets, Bulgaria, 2007), 378-382.

[13] Morinaga, S., Yamanishi, K., Tateishi, K., and Fukushima, T. 2002. Mining product reputations on the Web. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (KDD '02). ACM, New York, NY, USA, 341-349.

[14] Nivre, J. 2005. Dependency grammar and dependency parsing. *Technical Report MSI report* 05133, Växjö University: School of Mathematics and Systems Engineering

[15] O'Hare, N., Davy, M., Bermingham, A., Ferguson, P., Sheridan, P., Gurrin, C., and Smeaton, A.F. 2009. Topic-dependent sentiment analysis of financial blogs. In*Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion* (TSA '09). ACM, New York, NY, USA, 9-16.

[16] Pang, B. and Lee, L. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics* (Barcelona, Spain, 2004). Association for Computational Linguistics, 271-278.

[17] Quirk, R., Greenbaum, S., Leech, G., and Svartvik, J. 1985. *A Comprehensive Grammar of the English Language.* Longman

[18] Riloff, E. and Wiebe, J. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing - Volume 10*. Association for Computational Linguistics, 105-112.

[19] Shaikh, M. A. M., Prendinger, H. and Ishizuka, M. 2008. Sentiment Assessment of Text By Aanalyzing Linguistic Features And Contextual Valence Assignment. *Appl. Artif. Intell.*, 22, 6, 558-601.

[20] Tan, L.K.W., Na, J.-C., and Theng, Y.-L. 2011. Influence Detection between Blog Posts through Blog Features, Content Analysis, and Community Identity. *Online Information Review*, 35, 3.

[21] Thet, T. T., Na, J.-C. and Khoo, C. S. G. 2010. Aspect-Based Sentiment Analysis of Movie Reviews on Discussion Boards. Journal of Information Science, 36(6):823-848

[22] Turney, P. D. Thumbs up or thumbs down? 2002. Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics ACL* (2002), 417-424.

[23] Wilson, T., Wiebe, J. and Hoffmann, P. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing* (Vancouver, British Columbia, Canada, 2005). Association for Computational Linguistics, 347-354.

[24] Wilson, T., Wiebe, J. and Hwa, R. 2006. Recognizing Strong and Weak Opinion Clauses. *Computational Intelligence*, 22, 2, 73-99.

[25] Yi, J., Nasukawa, T., Bunescu, R. and Niblack, W. 2003. *Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques*. Institute of Electrical and Electronics Engineers Computer Society, 427-434.