Tan LKW, Na JC, Theng YL *et al.* Phrase-level sentiment polarity classification using rule-based typed dependencies and additional complex phrases consideration. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 27(3): 650–666 May 2012. DOI 10.1007/s11390-012-1251-y

Phrase-Level Sentiment Polarity Classification Using Rule-Based Typed Dependencies and Additional Complex Phrases Consideration

Luke Kien-Weng Tan[1] (陈坚永), Jin-Cheon Na[1] (罗镇川), *Member*, *ACM*, Yin-Leng Theng[1] (邓燕玲), and Kuiyu Chang[2] (张圭煜)

[1] *Wee Kim Wee School of Communication and Information, Nanyang Technological University, 31 Nanyang Link 637718, Singapore*

[2] *School of Computer Engineering, Nanyang Technological University, Block N4 Nanyang Avenue, 639798, Singapore*

E-mail: {w080078, tjcna, tyltheng, askychang}@ntu.edu.sg

Received September 1, 2011; revised January 19, 2012.

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. Providers of such content exert a certain level of influence on the receivers and this is evident from blog sites having effect on their readers' purchase decisions, political view points, financial planning, and others. 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, and in addition consider the effects of the more complex relationships found between words in sentiment polarity classification. 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 (CSRs) to automatically learn the typed dependency patterns, and benchmark the performance of CSR against a heuristic method. Preliminary results show CSR leads to further improvements in classification performance achieving over 80% F1 scores in the test cases. In addition, we observe more complex relationships between words that could influence phrase sentiment polarity, and further discuss on possible approaches to handle the effects of these complex relationships.

Keywords class sequential rule, complex phrase, sentiment analysis, typed dependency

1 Introduction

The growth and popularity of opinion-rich websites such as blogs and online forums 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. Providers of such content exert a certain level of influence on the receivers and this is evident from blog sites having effect on their readers' purchase decisions (e.g., www.engadget.com), political view points (e.g., www.huffingtonpost.com), financial planning (e.g., www.cashmoneylife.com), and others. The ability to detect influence in the blogosphere could identify the influential blogger and the chain of information flow. Through this, stimulus could be added to aid the flow of positive information, or pre-emptive and preventive actions taken to minimize any negative impact. The identification of the influential bloggers could be used in various applications. For example, influential bloggers are often market-movers where they can influence the buying decisions of fellow bloggers, and identifying them can help companies better understand the key concerns and new trends about the products, and provide the influential bloggers with additional information to turn them into unofficial spokesmen. Influential bloggers could also express opinions on government policies, which affects the reactions of the readers. Tapping on the influential bloggers can help understand the changing interests, foresee potential pitfalls and likely gains, and proactively adapt plans in a timely manner. The influential bloggers could also help in customer support and troubleshooting as their posted solutions are usually closely followed.

Regular Paper

Instead of going through every blog post, companies could start with the influential bloggers' posts to identify the issues.

Previous studies[1-2] linked information propagation and influence to blog features which are mainly graph-based, such as the number of in- and out-links. However, the use of blog features alone to detect influence in the blogosphere may not yield highly accurate results. This is because influence is a subjective concept and often depends on the context of the posting, which means that a deeper analysis on the post content is required to improve influence detection performance. It was further observed in the studies by Agarwal *et al.*[2] and Tan *et al.*[3] that acquiring the ability to detect sentiments expressed in the content could identify the chain of influence flow. 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. Further to that, we also consider the more complex relationships between words that could affect the sentiment polarity classification.

Sentiment analysis has been employed in various applications such as opinion-based search engines[4-5], where the ability to search for topics with extreme sentiments could help governmental institutions find terrorist organizations using the Web as a communication tool, automatic detection of sentiments of financial blogs[6-7] for market evaluation, including during a financial crisis, and identification of negative sentiments on companies or products[8-9] in a commercial crisis where a lapse in delivery of quality services and products have led to bad reputation.

Sentiment analysis had been studied at both document and sentence levels with the goal of assigning an overall sentiment polarity for the document or sentence[10-11]. However, Pang and Lee[12] highlighted that sentiment often could be expressed in a more subtle manner, making it difficult to be identified by any of a sentence or document's terms when considered in isolation. Other studies had proposed methods to predict sentiment polarity at the clause or phrase levels to provide a more refined analysis[13-15]. Previous linguistic approaches in sentiment analysis had leveraged on semantic dependencies between words to predict sentiments. Wilson *et al.*[15] observed that word patterns along with its modifiers could determine the intensity of phrase sentiments. For example, in the phrase "indiscriminate killing", the negative polarity 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[16]. 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 our study, we focus on deriving polarity pattern rules at the phrase level, which we also define as the bigrams in typed dependencies. We use the Stanford parser① to generate typed dependencies from our sentence dataset. Fig.1 shows examples of the adjectival modifier (AMOD), adverbial modifier (ADVMOD), and direct object modifier (DOBJ) typed dependencies for the respective phrases. The extracted patterns are then used in a linguistic approach to determine the sentiment polarity of unseen bigram phrases.

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

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

In the Stanford parser, an AMOD of a noun phrase is any adjectival phrase that serves to modify the meaning of the noun phrase, an ADVMOD of a word is an adverb or adverbial phrase that serves to modify the meaning of the word, and the DOBJ of a verb phrase is the object of the verb. Fig.2 shows a typed dependency tree generated for the sentence "The cruel dictator willfully suppressed the citizens". The DET is the determiner, which shows the relation between the head of the noun phrase and its determiner. The nominal subject (NSUBJ) is a noun phrase, which is the syntactic subject of a clause.

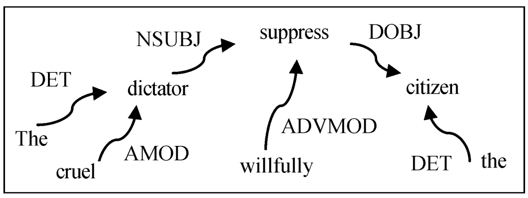

Fig.2. Typed dependency tree.

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

The corresponding typed dependency bigrams are listed in Fig.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.

DET(dictator, The)
AMOD(dictator, cruel)
NSUBJ(suppress, dictator)
ADVMOD(suppress, willfully)
DET(citizens, the)
DOBJ(suppress, citizens)

Fig.3. Generated typed dependencies.

From Fig.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, cruel)" could indicate a polarity pattern rule of "AMOD(NN(−),JJ(−)) → (−)" giving a negative output, where NN(−) and JJ(−) refer to a negative noun term and a negative 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[15,17]. 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 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.*[18] 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.*[13] 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 the manual heuristic approach, our proposed approach could automatically generate a comprehensive set of rules to provide the sentiment classification. In our study, we extracted the typed dependency bigrams of the sentences using the Stanford parser and apply Class Sequential Rules (CSRs) proposed by Liu[19] to derive polarity class rules from sequential patterns within the bigrams. We then compare the CSRs with the heuristic rules adapted from Thet *et al.*[13] using the dataset from Pang and Lee[12] for our experiments. 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 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 of complex phrases and conclusions.

2 Related Work

Previous studies[10-11,20] have studied sentiment analysis at the document and sentence levels to predict the overall sentiment polarity for the document or sentence. Polarity could be in the form of positive, negative, or at times neutral. Osman *et al.*[20] proposed a document level opinion detection system that can find more relevant opinion documents using five system fusion methods, that is, a voting method, an inverse rank method, a linear-normalized score method, and two weighted methods. The resulting document list can then be used as a list for a search engine or as input into an opinion analysis system for further analysis. Hu and Liu[21] mined and summarized customer reviews of electronic products, such as digital cameras, cellular phones, and mp3 players. They extracted the features or aspects (such as picture quality and screen size) of the product on which the customers have expressed their opinions, and predicted whether each opinion sentence is positive or negative. If positive or negative opinion words prevail, the opinion sentence is predicted as a positive or negative one. The approach used by Kim and Hovy[22] first identifies an opinion bearing word, it then labels semantic roles related to the word in the sentence, and lastly finds a holder and a topic of the opinion word among labeled semantic roles. Zhang *et al.*[23] studied sentiment analysis of Chinese content rather than English. They used a rule-based approach to determine a sentence's sentiment based on word dependency, and then predicted a document sentiment by aggregating the analysis results of individual sentences. Sentiment analysis is not without challenges. Sentiments could be expressed in a subtle manner, making it difficult to be identified at the sentence or document level[12]. Sentiment and subjectivity are context-sensitive and, at a coarser granularity, domain or document genre dependent[24]. Further to that, the order in which different opinions are presented can result in a completely opposite overall sentiment polarity. These considerations have complicated sentiment analysis at the document and sentence level. Other studies had taken a different approach to analyze the finer-level relationships between words to predict the sentiment polarity output at the sentence or phrase level.

More recent sentiment analysis studies had used a linguistic approach which leveraged on the semantic dependencies between words to predict sentiments. Wilson *et al.*[15] 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 appear in a specific pattern involving a word and just one of its modifiers were included as features in the evaluation. A common approach to extracting patterns within word dependencies for sentiment detection was through heuristic means[13,18,25]. Moilanen and Pulman[25] proposed a sentiment composition model based on the concept that the global polarity of a sentence 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 ($\mathrm{OUT}^{\alpha_{ij}} \rightarrow \mathrm{SPR}^{\alpha_i} + \mathrm{SUB}^{\alpha_j}$) 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.*[18] 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 $[\mathrm{ADJ}_{\mathrm{pos}} + (\mathrm{CON}_{\mathrm{neg}}|\mathrm{NE}_{\mathrm{neg}}) \rightarrow$ (negative output)] (e.g., "strong cyclone") and $[\mathrm{ADJ}_{\mathrm{pos}} + (\mathrm{CON}_{\mathrm{pos}}|\mathrm{NE}_{\mathrm{pos}}) \rightarrow$ (positive output)] (e.g., "brand new car") where the adjectives (ADJ), concepts (CON), and named entities (NE) were assigned valence groupings.

Thet *et al.*[13] 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 score 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 $[(\mathrm{ADJ}_{\mathrm{pos}} + \mathrm{N}_{\mathrm{pos}}) \rightarrow$ (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. Furthermore, the polarity pattern rules could be generalized across different domains.

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 parsed 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 CSRs with the heuristic rules (HRs) adapted from Thet *et al.*[13]. We employed the CSR technique in our study to mine the subsequence patterns of both unigrams (e.g., "NN(+), _", where "_" represents a term with any possible POS and polarity) and bigrams (e.g., "NN(+), JJ(+)") patterns because the sequence (i.e., ordering) of the features is taken into consideration while generating the typed dependencies rules. Fig.4 gives an overview of our CSR and HR approach.

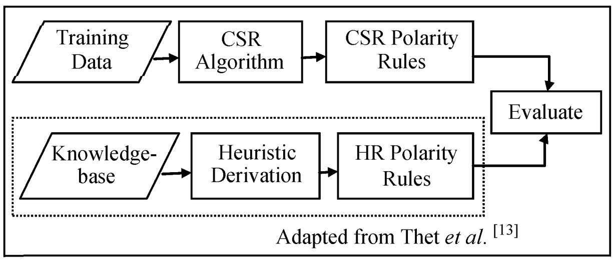

Fig.4. Overview of the CSR and HR approaches.

3.1 Dataset

We used a readily available subjectivity dataset from Pang and Lee[12] to obtain 1 000 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. Next, we tagged the polarity of each word in the typed dependencies bigrams by matching the words to the subjectivity lexicon terms from Thet *et al.*[13], which were derived from SentiWordNet② and the subjectivity lexicon of Wilson *et al.*[14] Each word in the typed dependencies bigrams is tagged one of three polarity values, 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) tags. It is observed that there are generally more negative terms than positive terms, except for the adverbs. 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[26] value of 0.78, indicating an acceptable reliability of the coded polarities.

Table 1. Polarity Distribution of the Subjectivity Lexicon Terms

POS	Positive (+)	Negative (−)	Neutral (0)
Adjectives	5 375	8 392	1 027
Adverbs	2 010	934	305
Verbs	1 149	2 303	714
Nouns	5 407	8 591	1 409

The respective typed dependency bigrams' polarity distributions in the coded dataset are shown in Table 2.

Table 2. Polarity Distribution of Bigrams in Coded Dataset

Type	Positive (+) Class	Negative (−) Class	Neutral (0) Class	Total
AMOD	320	228	452	1 000
ADVMOD	245	222	533	1 000
DOBJ	196	153	641	1 000

It is observed for the dataset that more opinions are expressed using adjectival words as seen from AMOD having the most sentiment polarized phrases, while DOBJ has the least number. Fig.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 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+) → (−)

Fig.5. Sample generated output of coded data.

Wilson *et al.*[15] found that the sparsity of word occurrences in clauses posed a problem in extracting meaningful patterns. Joshi and Penstein-Rosé[27] 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. Using the generalized POS-polarity dataset, we derive the POS-polarity pattern rules through the semi-supervised CSR algorithm. Fig.6 shows samples of the POS-polarity pattern rules, in the form of "typed-dependency(governor-POS[polarity], dependent-POS[polarity]) → (bigram-polarity)". We further evaluated the effects of generalization by reducing the features to just the polarity of each word in the bigram. Fig.7 shows samples of the polarity-only pattern rules, indicating the relation "typed-dependency(governor[polarity], dependent[polarity]) → (bigram-polarity)". Our results show that the performance of the polarity-only pattern rules is on-par to that of the POS-polarity pattern rules.

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

Fig.6. Sample POS-polarity pattern rules.

AMOD((0), (+)) → (+)
ADVMOD((+), (+)) → (+)
DOBJ((−), (+)) → (−)

Fig.7. Sample polarity-only pattern rules.

At the phrase level, that is, 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 terms, which can be domain dependent, and the part-of-speech, and considering polarity as the only feature in the polarity evaluation, we further generalized our model across different domains. Sentiment scores which could have provided indication of the varying degrees of sentiment

②http://sentiwordnet.isti.cnr.it.

intensity are not used in our evaluation due to lack of records in our dataset to produce significant enough score-based polarity pattern rules. However, sentiment scores could be explored in future work on a larger dataset.

3.2 Class Sequential Rules (CSRs)

Liu[19] introduced Class Sequential Rules (CSRs), which is similar to Association Rule Mining[28-29] by using the concept of frequent item set except CSR considers the subsequence (called sequential pattern) and class labels of individual records. We adapted Liu's[19] 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 extracted both unigram and bigram pattern rules based on an experimentally determined minimum support threshold of 0.01 as suggested by Hu and Liu[21]. Additionally, we used 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, defined as follows.

$$\text{F1} = \frac{2 \times \text{Support} \times \text{Confidence}}{\text{Support} + \text{Confidence}}.$$

We validated the confidence threshold values by standardizing the support threshold value to 0.01 and varying the minimum confidence threshold value from 0.1 to 0.7. It was observed that beyond a confidence threshold value of 0.7, there were no valid polarity pattern rules generated for certain test records. The average F1 scores of the respective confidence threshold values and the average, minimum, and maximum number of pattern rules generated for the 10-fold cross validation tests are shown in Table 3. From the results of Table 3, there is no significant difference in average F1 scores among the tested confidence values at the $p < 0.05$ level for the conditions $[\text{F}(6, 63) = 0.007, p = 1.000]$ based on the one-way ANOVA testing. The results further show that the number of generated patterns is the greatest when the confidence value is 0.1.

Table 3. Results for Respective Confidence Threshold Values

Confidence Value	Avg. F1 (%)	Avg. No. Pattern	Min. No. Pattern	Max. No. Pattern
0.1	84.45	38.6	36	41
0.2	84.71	29.4	27	32
0.3	84.39	27.4	25	29
0.4	84.39	26.3	24	28
0.5	84.39	24.4	22	26
0.6	84.39	23.8	22	26
0.7	84.39	22.3	20	24

Note: support threshold value = 0.01.

We further validated the support threshold values by standardizing the confidence threshold value using the arbitrary chosen value of 0.5 and varying the minimum support threshold value.

Table 4 shows the average F1 scores of the respective support threshold values and the average, minimum, and maximum number of generated pattern rules for the 10-fold cross validation tests. From the results of Table 4, there is no significant difference in average F1 scores among the tested support values at the $p < 0.05$ level for the conditions $[\text{F}(9, 90) = 0.183, p = 0.995]$ among the respective support threshold values based on the 1-way ANOVA testing. The results also show that the number of generated patterns is the greatest when the support value is 0.01. From the validation tests, we used the support threshold value of 0.01, which provides the highest average F1 score, and an arbitrarily chosen confidence threshold value of 0.5 for our subsequent experiment testing.

Table 4. Results for Respective Support Threshold Values

Support Value	Avg. F1 (%)	Avg. No. Pattern	Min. No. Pattern	Max. No. Pattern
0.01	84.97	24.4	22	26
0.02	84.50	16.4	15	17
0.03	83.93	13.5	13	14
0.04	83.35	12.5	11	13
0.05	83.04	10.4	10	11
0.06	83.03	10.0	10	10
0.07	83.04	10.0	10	10
0.08	83.04	9.8	9	10
0.09	83.04	9.5	9	10
0.10	83.60	9.0	9	9

Note: confidence threshold value = 0.5.

3.3 Heuristic Rules

Although Thet *et al.*[13] studied sentence-level sentiment polarity prediction, we adapted their phrase-level heuristic rules implementation as our baseline because our focus is at the phrase-level. Specifically, we used only three typed dependencies rules (i.e., AMOD, ADVMOD, and DOBJ) for comparison. The heuristic rules (HR) used in our study are listed in Table 5. We further separated the neutral class rules, which were assumed positive in Thet *et al.*[13], to make a direct comparison with the three CSR classes. Sentiment scores were used in Thet *et al.*[13], which we did not consider due to the limited number of records in the dataset to produce significant score-based polarity pattern rules. Negation was considered in the heuristic rules in Thet *et al.*[13] to improve the sentiment scoring of 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", negation of the negative verb "fail" by the negating term "hardly" would make the phrase positive.

Table 5. Heuristic POS-Polarity Pattern Rules (HR)

HR Type	Positive Class Patterns	Negative Class Patterns	Neutral Class Patterns
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

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.

4 Evaluation

10-fold cross validation for each of the positive (+), negative (−), and neutral (0) classes 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.*[13] are coded into the evaluation method with each record processed by the heuristic rules. The performance of the CSRs are then compared with the modified heuristic method adapted from Thet *et al.*[13]. We compare in the following subsections each of the three major typed dependencies for CSR and HR.

4.1 AMOD Evaluation

The top 10 generated AMOD CSRs with respect to their F1 scores are listed in Table 6, while the list of generated AMOD CSRs are shown in Table 7.

Table 6. Top 10 AMOD CSR POS-Polarity Rules

No.	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)	+

Table 7. CSR POS-Polarity Rules for AMOD Typed Dependency: AMOD(governor, dependent)

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

The rules are in the order of "governor-POS (polarity), dependent-POS (polarity)", with the unigram pattern rules represented as "governor-POS (polarity), _" and "_, dependent-POS (polarity)", where "_" represents a term with any possible POS and polarity. In comparison with the HR patterns of Table 5, 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 additionally 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.

From Table 8, where the polarity-only pattern rules are shown, the HR AMOD(−, +) → (−) pattern is covered in CSR by the AMOD(−, _) → (−) unigram rule.

Table 8. AMOD Polarity-Only Rules

Sequence Type	CSR Patterns (+)	CSR Patterns (−)	CSR Patterns (0)	HR Patterns (+)	HR Patterns (−)	HR Patterns (0)
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

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 the CSR polarity method, when compared to HR as shown in Table 9. 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 classifies the bigram as positive (+) based on the polarity of the dependent term "biggest".

As seen in Table 9, the F1 values for the four methods, POS-polarity and polarity features for both CSR and HR approaches 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 9, 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 9. Results for AMOD POS-Polarity and Polarity-Only Rules

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

4.2 ADVMOD Evaluation

The top 10 generated ADVMOD CSR pattern rules with respect to their F1 scores are given in Table 10, with the generated ADVMOD CSR pattern rules listed in Table 11.

Table 10. Top 10 ADVMOD CSR POS-Polarity Rules

No.	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)	_	−

The generated ADVMOD polarity rules are shown in Table 12. 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.

Table 11. CSR POS-Polarity 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)	• JJ−, RB+ (intrusive, simply) • JJ−, RB (flashy, visually) • JJ− RB− (dull, unspeakably)	• VB, RB (find, only) • VB, RB+ (cut, just) • NN, RB+ (comparison, much)
• VB+, RB+ (delighted, undoubtedly) • VB+, RB (engrossing, so) • NN+, RB+ (fun, simply)	• 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)	• 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 12. ADVMOD Polarity-Only Rules

Sequence Type	CSR Patterns (+)	CSR Patterns (−)	CSR Patterns (0)	HR Patterns (+)	HR Patterns (−)	HR Patterns (0)
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

From Table 13, 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) type records, which are not discovered in HR.

Our implementation classifies all unmatched bigrams into the neutral (0) class, which explains the high recall and low precision (54.49%) values for the neutral (0) class in the HR POS-polarity method. Thet *et al.*[13], 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 13. Results for ADVMOD POS-Polarity and Polarity-Only Rules

Method	Class	Precision (%)	Recall (%)	F1 (%)	Avg. F1 (%)
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-only)	(+)	75.06	80.94	77.47	83.10
	(−)	80.75	98.15	88.38	
	(0)	89.58	78.34	83.45	
HR (polarity-only)	(+)	52.53	97.51	67.93	74.34
	(−)	80.75	98.15	88.38	
	(0)	98.98	50.70	66.73	

4.3 DOBJ Evaluation

The top 10 DOBJ CSR pattern rules are given in Table 14, while the CSR generated pattern rules are listed in Table 15. The DOBJ polarity rules are similar except for the CSR unigram rules and the HR$(+,0) \rightarrow (+)$ rule, which was not discovered in CSR method as shown in Table 16.

Table 14. Top 10 DOBJ CSR POS-Polarity Rules

No.	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 confidence of $(\_,0) \rightarrow (+)$ is at 0.13, which is less than the minimum threshold of 0.5, and was not significant enough for the CSR$(+,0) \rightarrow (+)$ rule to be generated, and the pattern for $(+,0) \rightarrow (+)$ was covered using the unigram CSR$(+,_) \rightarrow (+)$ 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 as seen in Table 17. This is because the neutral (0) class unigram rule CSR$(_,0) \rightarrow (0)$ has a higher F1 $= 0.79$ value, which overrides the positive unigram rule CSR$(+,_) \rightarrow (+)$ of

Table 15. CSR POS-Polarity 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 16. DOBJ Polarity-Only Rules

Sequence Type	CSR Patterns (+)	CSR Patterns (−)	CSR Patterns (0)	HR Patterns (+)	HR Patterns (−)	HR Patterns (0)
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

F1 = 0.68 in classifying (+, 0) patterns, which shows the dependency of CSR on the dataset polarity frequency.

The results for CSR POS-polarity and HR POS-polarity method are similar as seen from Table 17. 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 (Avg. F1 = 80.95%) and heuristic rules method (Avg. F1 = 79.54%) are not significantly different. From Table 17, 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.

Table 17. Results for DOBJ POS-Polarity and Polarity-Only Rules

Method	Class	Precision (%)	Recall (%)	F1 (%)	Avg. F1 (%)
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-only)	(+)	75.18	56.20	63.72	81.45
	(−)	85.56	100.00	92.09	
	(0)	86.87	90.34	88.54	
HR (polarity-only)	(+)	70.24	99.56	82.22	88.09
	(−)	85.42	98.85	91.50	
	(0)	99.62	83.08	90.57	

5 Interpretation of CSR Results

In this study, we explored the use of the CSR method to derive the polarity predicting pattern rules and compared the results between the CSRs 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 CSRs, 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 in 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. 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, an AMOD$((-),(-))$ polarity pattern rule will give the same negative output regardless of the domain. Possible causes of error in the tests are the wrongly parsed typed dependencies such as AMOD (regard:VB, be:VB), and words from the subjectivity lexicon which were tagged with a wrong polarity. Errors in the typed dependency parsing hold 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. It was observed that there exist more complex relationships between words that could lead to polarity conflict cases, which were systematically processed according to the priority of the generated rules in our study. To handle the observed complexity found between words, we discuss possible approaches in the next section.

6 Consideration for Complex Phrases

6.1 Complex Relationships Between Words

Though our results showed that polarity pattern rules using generalized features performed well, a semantic analysis of the words is still required to identify the more complex relationships between words that could influence phrase polarity. It was observed in Wilson *et al.*[14] and Polanyi and Zaenen[30] that there exist more complex relationships between words that could influence the sentiment polarity of phrases. Examples of these complex relationships include negation (e.g., "not" in "not good"), subjective terms found in domain terms (e.g., "star" in "star trek"), subjective terms appearing as neutral (e.g., "fiction" in "science fiction"), neutral terms appearing as subjective (e.g., "red" in "red carpet"), and intensifier and mitigation terms.

Intensifiers such as "very" intensify the sentiments of their adjacent terms (e.g., "very good"), while mitigators such as "few" reduce the sentiment (e.g., "few support"). Wilson *et al.*[14] considered a word to be an intensifier if it appears in a list of intensifiers and if it precedes a word of the appropriate part of speech (e.g., an intensifier adjective must come before a noun). The paper compiled intensifier words from those listed in Quirk *et al.*[31], intensifiers identified from existing entries in the subjectivity lexicon, and intensifiers identified during exploration of the developmental data. Quirk *et al.*[31] 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). Polanyi and Zaenen[30] observed that there are contextual valence shifters that could influence the sentiment polarity of sentences. These include negatives (e.g., "is not"), intensifiers (e.g., "deeply"), presuppositional items (e.g., "barely"), modals (e.g., "might"), and irony. Negation can be local (e.g., "not good"), or involve longer distance dependencies such as the negation of the proposition (e.g., "does not look very good"). Simple negatives include "never", "none", "nobody", "nowhere", "nothing", "neither", etc. Polanyi and Zaenen[30] further observed that modal operators set up a context of possibility or necessity and in texts they initiate a context in which valence terms express an attitude towards entities which do not necessarily reflect the author's attitude towards those entities in an actual situation under discussion.

Although previous studies[14,30] have observed more types of complexities between words, our study focus only on the complex relationships found between words in a phrase. These complex relationships in a phrase are not easily detected using polarity patterns rules, and hence further analysis on the semantic meaning of words in a phrase is required to improve the phrase-level sentiment polarity classification performance. For example, subjective words could be found in a neutral sentiment polarity phrase such as the negative word "close" in the neutral phrase "close up" as in "take a close up look".

The sentiment polarity of words in a phrase could also be reversed, for example the negative word "break" in the positive phrase "break through". However, as typed dependencies contain only the grammatical relationship between two words, they could not detect the subjectivity within multiple words in a phrase. For example, in the sentence "The ra-ra skirts were all the rage in the 1980s", typed dependency analysis alone could not detect the positive expression "all the rage." as shown in Fig.8. A multi-word analysis is required to identify the subjective phrases. A copula (COP) is the relation between the complement of a copular verb and the copular verb. A predeterminer (PREDET) is the relation between the head of a noun phrase and a word that precedes and modifies the meaning of the noun phrase determiner. A prepositional modifier (PREP) of a verb, adjective, or noun is any prepositional phrase that serves to modify the meaning of the verb, adjective, noun, or even another preposition.

DET(skirts-3, The-1)
AMOD(skirts-3, ra-ra-2)
NSUBJ(rage-7, skirts-3)
COP(rage-7, were-4)
PREDET(rage-7, **all**-5)
DET(**rage-7, the**-6)
PREP(rage-7, in-8)
DET(1980s-10, the-9)
POBJ(in-8, 1980s-10)

Fig.8. Subjective phrase across typed dependencies.

In previous studies, Wilson *et al.*[14] used a machine-learning approach to predict the phrase sentiment polarity by identifying the appropriate subjective phrase features, while Moilanen and Pulman[25] analyzed the words constituent within phrases to detect sentiment polarity. In this paper, we further discuss the use of a linguistic approach in considering the more complex relationships between words, including subjective phrases, to improve the performance of phrase-level sentiment polarity classification as well as to provide a clear explanation of the grammatical relationships that gives the sentiment polarity.

6.2 Subjective Phrase Detection

Multi-word subjective phrases that influence sentiment polarity output could be found in sentences. For example, the sentence "The team that brought to life the worlds of Avatar is breaking new ground" contains the positive phrases "brought to life" and "breaking new ground", which cause the overall sentiment polarity output to be positive. We define subjective phrases as phrases that contain sentiment polarity orientation. A relative clause modifier (RCMOD) of an noun phrase is a relative clause modifying the noun phrase. An auxiliary (AUX) of a clause is a non-main verb of the clause.

As can be seen from Fig.9, using the typed dependency polarity pattern rules that relied on the grammatical relationships, and the polarity patterns between two words is not sufficient to determine the subjectivity found within the multi-word expressions, which are otherwise neutral terms on their own. For example, using the DOBJ(*breaking*(0), *ground*(0)) $\rightarrow$ (0) and AMOD(*ground*(0), *new*(0)) $\rightarrow$ (0) typed dependencies polarity pattern rules would have resulted in an incorrect neutral polarity output for the positive phrase "breaking new ground". On the other hand, as seen from Fig.10, by extracting and analyzing the verb phrases "brought to life" and "breaking new ground" for subjectivity, we could correctly detect the positive sentiment polarity expressed by the phrase.

DET(team, The)
NSUBJ(breaking, team)
NSUBJ(brought, that)
RCMOD(team, **brought**)
PREP(brought, **to**)
POBJ(to, **life**)
DET(worlds, the)
DOBJ(brought, worlds)
PREP(worlds, of)
POBJ(of, Avatar)
AUX(breaking, is)
AMOD(ground, **new**)
DOBJ(**breaking**, **ground**)

Fig.9. Generated typed dependencies.

Hence, we have to evaluate the expressions within phrase entities to accurately predict the phrase subjectivity. Examples of phrase entities include adverbial phrase (ADVP), noun phrase (NP), verb phrase (VP).

In order to handle the subjective phrases that could influence the sentiment polarity output, we could leverage on prior knowledge base to detect subjectivity in the multi-word expressions to identify phrase subjectivity and adjust the output polarity accordingly. For

```
((ROOT
 (S
  (NP
   (NP (DT The) (NN team))
   (SBAR
    (WHNP (WDT that))
    (S
     (VP (VBN brought)
      (PP (TO to)
       (NP (NN life)))
      (NP
       (NP (DT the) (NNS worlds))
       (PP (IN of)
        (NP (NNP Avatar))))))))
   (VP (VBZ is)
    (VP (VBG breaking)
     (NP (JJ new) NN ground))))))
```

Fig.10. Generated phrase structure tree.

example, we could assign a positive prior polarity tag to the phrase "feel at home", and a negative prior polarity tag to the phrase "red handed". This subjectivity phrase lexicon containing phrase-lexicon terms could be built from extracting subjective phrases from the training dataset and other knowledge base containing phrase information (e.g., phrasal verbs dictionary and idioms dictionary), and humans can manually annotate the phrases with the mutually agreed sentiment polarity. Examples of subjective phrase terms are shown in Table 18.

Table 18. List of Sample Subjective Phrases

Polarity	Example Phrases
Positive	red carpet, above par, cutting edge, stand the test
Negative	set you back, arm and a leg, wet blanket, wear out
Neutral	an hour long, up close, big apple, dead even

In the steps to evaluate phrase subjectivity, we would first process the earlier proposed CSR typed dependencies polarity pattern rules before processing the subjective phrase polarity rules. In order to relate the typed dependencies to their respective phrase entity, there is a need to analyze the phrase structure tree. We could group the typed dependencies to be evaluated under the corresponding phrase according to the phrase head word (= H). For example, from Fig.11 in the sentence "he will get into hot water with the director", the typed dependencies with "get" as their governor term are identified with the verb phrase "VP (VB = H get)". This is similarly done for the governor terms "water" and "director" with their corresponding phrase entities.

From Fig.12, focusing on the negative expression "get into hot water", we evaluate the lower nested level noun phrase "(NP (JJ **hot**) (NN=**H water**))" using the AMOD(***water***(0), ***hot***(+)) → (+) typed dependency pattern rule to give an interim positive output. Next, we would then use a bottom-up approach in evaluating the rules, taking into consideration the nested level of each item in the phrase structure tree to ensure that only items from the same level are evaluated and the resultant polarity propagated to subsequent levels for further evaluation. The positive AMOD output is recursively evaluated with the corresponding neutral prepositional (PP) typed dependencies and the AUX typed dependency "AUX(*get*, *will*)" to obtain the verb phrase "will get into hot water", and we match the verb phrase with the terms from the subjective phrase lexicon to get the negative polarity output for the phrase "get into hot water".

```
(VP (VB = H get)
     nsubj(get, he)
     aux(get, will)
     prep_into(get, water)
     prep_with(get, director)
(NP (JJ hot) (NN=H water)))
     amod(water, hot)
(NP (DT the) (NN=H director)))))))
     det(director, the)
```

Fig.11. Phrase entities with the corresponding typed dependencies.

```
(ROOT
 (S=H
  (NP (PRP=H he))
  (VP=H (MD=H will)
   (VP (VB=H get)
    (PP (IN=H into)
     (NP (JJ hot)(NN=H water)))
     (PP (IN=H with)
      (NP (DT the) (NN=H director)))))))
```

Fig.12. Generated phrase structure tree.

We leveraged on the phrase structure tree to identify the nested level of iteration items to ensure that only same-level items are evaluated before moving on to higher level items. The typed dependency tree is not used in our approach as it is harder to identify the phrase structure from a typed dependency tree, as shown in Fig.13, compared with the phrase structure tree.

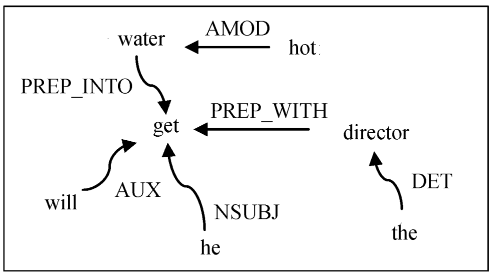

Fig.13. Typed dependency tree.

6.3 Phrase Polarity Scores and Intensity Rules

The intensity of sentiment polarity affects the eventual output polarity. Therefore, we could use a polarity score value ranging from -1 to $+1$ to measure the intensity of sentiment polarity. Examples of the prior sentiment scores assigned to words are shown in Table 19.

Table 19. List of Prior Sentiment Scores of Terms

Category	Score	Example Terms
High Negative	-1	conspire, abuse
Intermediate Negative	-0.5	messy, defect
Neutral	0	obvious, thought
Intermediate Positive	0.5	unique, impress
High Positive	1	graceful, righteous

Due to the subjective nature of sentiment polarity, instead of using a continuous score value, the prior priority scores could be assigned discretized scores within the values $[-1, -0.5, 0, 0.5, 1]$ to generalize the sentiment polarity classification of words into the various categories of sentiment polarity intensity. It is also possible to assign sentiment scores to phrases as an indication of the phrase sentiment polarity intensity. For example, the high intensity value of "1" could be given to the positive phrase "red carpet", and the intermediate intensity value of "0.5" is given to the positive phrase "above par".

To handle the more complex relationships between words, additional lexicon-based intensity rules could be used. The idea is to identify key expressions that could alter the level of phrase intensity. The additional lexicons include negators, intensifiers, mitigators, maximizers, and minimizers. Examples of the lexicon terms and the intensity rules are shown in Tables 20 and 21 respectively.

Table 20. List of Sample Lexicon Terms

Lexicon	Example Terms
Negators	rather, never, unlikely
Intensifiers	widely, a lot, most
Mitigators	possibly, quite, somewhat
Maximizers	highly, absolutely, best
Minimizers	little, scarcely, a bit

Table 21. List of Intensity Rules

(M)odifier	(T)erm	Polarity Score	Examples
Negation	T	$-1 \times \mathrm{T}$	Hardly good
Intensify	T	$(2 \times \mathrm{T})$, max value $= \pm 1$	Highly recommended
Mitigate	T	$(0.5 \times \mathrm{T})$	Slightly better
Maximize	T	$1\times$ Polarity(T)	Totally bad
Minimize	T	$0.25 \times \mathrm{T}$	Least effort

The lexicon-based intensity rules evaluate the effects of each complex term and adjust the sentiment polarity output intensity accordingly. For the negation rule in Table 21, we reverse the polarity score of the word that is modified by the negator. In the intensify rule, the polarity score of the modified word is doubled, but limited to a value of ± 1 in consideration of the increasing effect of the intensifier. Conversely, the polarity of the modified word is halved in the mitigate rule. The polarity score is maximized to ± 1 in the maximize rule to denote the upper extreme of the intensity scale. In the minimize rule, the polarity score of the modified word is severely reduced to account for the lower extreme of the intensity scale. As an example of the intensity rule, in the sentence, "The movie is quite good", the mitigating effect of the word "quite" reduces the intensity, and hence the polarity score of the positive word "good" has a less positive sentiment polarity output. The typed dependency pattern rule is given as ADVMOD(*good*(0.5), *quite*(mitigate) $\rightarrow$ 0.25). In comparison, the simple ADVMOD(*good*(+), *quite*(0) $\rightarrow$ (+)) typed dependency polarity rule would not be able to detect the decrease in the positive sentiment intensity.

We further observed that subjective phrases not only determine the sentiment polarity of phrases, but could also alter the intensity of sentiment polarity through intensification (e.g., "long since" as in "he has long since won the award"), maximization (e.g., "the one and only"), mitigation (e.g., "more or less"), and minimization (e.g., "not going anywhere" as in "the film is not going anywhere"). We could collect and differentiate these intensity altering phrases from the polarity subjective phrases. The quality of the subjective phrase lexicons could also be improved by considering more interesting phrases that contain neutral words that are subjective in a phrase, subjective words that are neutral in a phrase, or subjective words that reverse the polarity in a phrase. For example, the negative phrase "give up" contains words that are positive when considered individually.

Other typed dependencies could be analyzed for use to detect the sentiment polarity of the phrase. Examples include phrase verb particle (PRT) and auxiliary (AUX). The phrase verb particle relation identifies a phrasal verb, and holds between the verb and its particle. For example, the PRT(*rip*, *off*) identifies and extracts the phrase "rip off" which is negative in sentiment polarity. An auxiliary of a clause is a non-main verb of the clause and could be used to detect intensity of the verb. For example, aux(*rid*, *should*) has a higher intensity than aux(*rid*, *has*). Nonetheless, the typed dependencies could only handle simple subjective phrases, while multi-word subjective phrases have to be analyzed separately.

6.4 Possible Issues

It is observed that there could be situations where typed dependency evaluation without considering the more complex relationships between words would still lead to the correct prediction. For example, in the positive sentiment orientated sentence, "The fans hope (+) that the producer will not (negation) call it a day (−) for the drama", which contains a positive verb, a negation, and a negative phrase. The typed dependency pattern rules could correctly predict a positive polarity output by evaluating the positive verb "hope" even without considering the negative phrase negation. This is because negation of the negative phrase would still result in a positive sentiment polarity output. Nevertheless, our proposed approach could provide a more comprehensive and a finer-grain level analysis in predicting the output, which is shown in a more accurate and higher positive polarity intensity score by considering the negation of the negative phrase.

It is also possible to provide a wrong sentiment polarity prediction due to the use of polarity phrases in different context. For example, the sentence "As the screws tighten, the tensions mount in the situation" contains the negative subjective phrase "screws tighten", which however would be interpreted as neutral in the sentence "The screws tighten the hinge to the door". One possible approach to handle the phrase contextual issue is to further consider the syntactic structure of the subjective phrase within the sentence. From Fig.14, it is observed that the subjective phrase from the first sentence is contained within a subject phrase (that is, the subject performing an action). While, the same phrase from the second sentence is part of a predicate (that is, a verb acting on an object) as seen in Fig.15. We could assign the negative sentiment polarity to the phrase only if it is contained within a subject phrase, and not otherwise.

```
(ROOT
 (S
  (SBAR (IN as)
  (S
   (NP (DT the) (NNS screws))
   (VP (VBP tighten))))
  (, ,)
  (NP (DT the) (NNS tensions))
  (VP (VBP mount)
   (PP (IN in)
    (NP (DT the) (NN court) (NN case))))))
```

Fig.14. Phrase structure tree of the first sentence.

```
(ROOT
 (S
  (NP (DT the) (NNS screws))
  (VP (VBP tighten)
    (NP (DT the) (NN hinge))
   (PP (TO to)
    (NP (DT the) (NN door))))))
```

Fig.15. Phrase structure tree of the second sentence.

Hence, it may be necessary to provide the additional phrase syntactic structure information to correctly predict the sentiment polarity of the subjective phrase with respect to its context. However, further work needs to be done to determine the syntactic structure in relation to the subjective phrases in different contexts.

From the study, our future work will focus on studying methods in handling the complex phrases to improve sentiment prediction performance. We hope to achieve this through detecting the subjective phrases and apply the appropriate rules to handle the phrase subjectivity that could affect overall sentiment polarity. A preliminary study conducted in Tan *et al.*[32] revealed that the use of typed dependencies polarity rules and the consideration for the complex relationships between words improve sentiment polarity classification at sentence-level when compared with the baseline of counting the number of positive and negative terms in the sentence.

7 Conclusions

As influence is a subjective concept, detecting influence flow within the blogosphere requires further analysis on the content to improve performance. Our study focuses on the automatic generation of rules to provide the sentiment polarity classification of phrases, which could be used to provide an in-depth sentiment analysis of the content. We evaluated the automatic generation of typed dependency rules and discussed the use of typed dependency rules for predicting phrase-level sentiment polarity. Though performance for the CSR generated polarity pattern rules is good, detailed error analysis reveals the existence of polarity conflicts that cannot be resolved by polarity patterns rules. These polarity conflicts arise due to the possible lexical relationships between words. It was further observed that there exist more complex relationships between words that could affect the sentiment polarity of sentences. Therefore, we additionally analyzed these complex relationships between words in an effort to improve the sentiment polarity classification performance. Our analysis takes into account the possible changes to the intensity of the sentiment polarity by considering intensifiers, maximisers, mitigators, minimisers, and negators. We further considered the subjectivity of phrases that could also affect the sentiment polarity output. We hope that by considering the more complex relationships between words found within a phrase, we could conduct a more in-depth analysis on the possible phrase-level influences that would affect overall sentiment polarity. However, detecting subjective phrases in context is a complex problem that warrants further investigation.

The key contributions of this paper are as follows. First, we proposed 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 studied 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. 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 POS. The results from this study would be useful for improving rule-based approaches to sentiment analysis. These sentiment polarity outputs of the bigrams can then be recursively evaluated to predict the sentence-level sentiment polarity. Further to that, we discussed possible methods in handling complex phrases, which could influence the overall sentiment polarity output.

Our study has provided a fine-grained analysis of word relationships, which has a major role in determining the sentiment polarity of a phrase. We believe that our study provides a sound basis for future work using typed dependencies to predict sentiment polarity.

References

[1] Adar E, Adamic L A. Tracking information epidemics in blogspace. In *Proc. Int. Conf. Web Intelligence*, Washington, DC, USA, Sept. 2005, pp.207-214.

[2] Agarwal N, Liu H, Tang L, Yu P S. Identifying the influential bloggers in a community. In *Proc. WSDM 2008*, New York, USA, Feb. 2008, pp.207-218.

[3] Tan L K W, Na J C, Theng Y L. Influence detection between blog posts through blog features, content analysis, and community identity. *Online Information Review*, 2011, 35(3): 425-442.

[4] Abbasi A, Chen H, Salem A. Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. *Trans. Inf. Syst.*, 2008, 26(3): Article No. 12.

[5] Demartini G, Siersdorfer S. Dear search engine: What's your opinion about...?: Sentiment analysis for semantic enrichment of web search results. In *Proc. SEMSEARCH 2010*, New York, USA, April 2010, Article No.4.

[6] Devitt A, Ahmad K. Sentiment polarity identification in financial news: A cohesion-based approach. In *Proc. ACL 2007*, Prague, Czech Republic, June 2007, pp.984-991.

[7] O'Hare N, Davy M, Bermingham A, Ferguson P, Sheridan P, Gurrin C, Smeaton A F. Topic-dependent sentiment analysis of financial blogs. In *Proc. CIKM Workshop on TSA 2009*, New York, USA, Nov. 2009, pp.9-16.

[8] Ding X, Liu B, Yu P S. A holistic lexicon-based approach to opinion mining. In *Proc. WSDM 2008*, New York, USA, April 2008, pp.231-240.

[9] Morinaga S, Yamanishi K, Tateishi K, Fukushima T. Mining product reputations on the Web. In *Proc. SIGKDD 2002*, New York, USA, July 2002, pp.341-349.

[10] Riloff E, Wiebe J. Learning extraction patterns for subjective expressions. In *Proc. EMNLP 2003*, Stroudsburg, PA, USA, July 2003, pp.105-112.

[11] Turney P D. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proc. ACL 2002*, Stroudsburg, PA, USA, July 2002, pp.417-424.

[12] Pang B, Lee L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proc. ACL 2004*, Barcelona, Spain, July 2004, pp.271-278.

[13] Thet T T, Na J C, Khoo C S G. Aspect-based sentiment analysis of movie reviews on discussion boards. *Journal of Information Science*, 2010, 36(6): 823-848.

[14] Wilson T, Wiebe J, Hoffmann P. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proc. HLT-EMNLP 2005*, Vancouver, British Columbia, Canada, Oct. 2005, pp.347-354.

[15] Wilson T, Wiebe J, Hwa R. Recognizing strong and weak opinion clauses. *Computational Intelligence*, 2006, 22(2): 73-99.

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

[17] Jakob N, Weber S H, Muller M C, Gurevych I. Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations. In *Proc. CIKM Workshop on TSA 2009*, Hong Kong, China, Nov. 2009, pp.57-64.

[18] Shaikh M A M, Prendinger H, Ishizuka M. Sentiment assessment of text by analyzing linguistic features and contextual valence assignment. *Appl. Artif. Intell.*, 2008, 22(6): 558-601.

[19] Liu B. Web Data Mining: Exploring Hyperlinks, Contents and Usage Data (1st edition). Springer Berlin Heidelberg, New York, 2006, pp.37-54.

[20] Osman D J, Yearwood J, Vamplew P. Weblogs for market research: Finding more relevant opinion documents using system fusion. *Online Information Review*, 2009, 33(5): 873-888.

[21] Hu M, Liu B. Mining and summarizing customer reviews. In *Proc. the 10th SIGKDD*, Seattle, WA, USA, Aug. 2004, pp.168-177.

[22] Kim S M, Hovy E. Determining the sentiment of opinions. In *Proc. the 20th COLING*, Geneva, Switzerland, 2004, pp.1367-1373.

[23] Zhang C, Zeng D, Li J, Wang F Y, Zuo W. Sentiment analysis of Chinese Documents: From sentence to document level. *Journal of the American Society for Information Science and Technology*, 2009, 60(12): 2474-2487.

[24] Na J C, Thet T T, Khoo C. Comparing sentiment expression in movie reviews from four online genres. *Online Information Review*, 2010, 34(2): 317-338.

[25] Moilanen K, Pulman S. Sentiment composition. In *Proc. RANLP 2007*, Borovets, Bulgaria, Sept. 2007, pp.378-382.

[26] Cohen J. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 1960, 20(1): 37-46.

[27] Joshi M, Penstein-Rose C. Generalizing dependency features for opinion mining. In *Proc. ACL-IJCNLP 2009*, Suntec, Singapore, Aug. 2009, pp.313-316.

[28] Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases. In *Proc. VLDB 1994*, Santiago de Chile, Chile, Sept. 1994, pp.487-499.

[29] Wong K W, Zhou S, Yang Q, Yeung J M S. Mining customer value: From association rules to direct marketing. *Data Mining and Knowledge Discovery*, 2005, 11(1): 57-79.

[30] Polanyi L, Zaenen A. Computing attitude and affect in text: Theory and applications. *Computing Attitude and Affect in Text: Theory and Applications*, 2006, 20: 1-10.

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

[32] Tan L K W, Na J C, Theng Y L, Chang K Y. Sentence-level sentiment polarity classification using a linguistic approach. In *Proc. ICADL 2011*, Beijing, China, Oct. 2011, pp.77-87.

Luke Kien-Weng Tan received the B.Eng degree from National University of Singapore (NUS), and the M.S. degree in information system from Nanyang Technological University (NTU), Singapore. Currently, he is a Ph.D. candidate in the Wee Kim Wee School of Communication & Information at NTU.

Jin-Cheon Na is an associate professor in the Wee Kim Wee School of Communication & Information at Nanyang Technological University (NTU), Singapore. Before he joined NTU, he was a senior researcher at the Agency for Defense Development, Korea. He obtained his Ph.D. degree from the Department of Computer Science at Texas A&M University. He has published more than 50 papers mainly in the areas of digital libraries, sentiment analysis, hypertext & hypermedia, document engineering, and knowledge organization. He is a member of the Editorial Review Board of the International Journal of Organizational and Collective Intelligence (IJOCI). He was also a non-executive director of DideoNet Co Ltd, Korea.

Yin-Leng Theng is an associate professor and associate chair (Research) in Division of Information Studies, Wee Kim Wee School of Communication and Information, Nanyang Technological University, Singapore. Her general research and teaching interests are in the areas of human-computer interaction; user interface design (interactive systems) — culture, ethics, security, privacy and trust; digital libraries; hypertext & web; e-learning & learning objects; education & information literacy and healthcare informatics.

Kuiyu Chang is an assistant professor of computer engineering at Nanyang Technological University, Singapore. Prior to that, he served as senior risk management analyst for ClearCommerce (now eFunds). From 2000 to 2002 he was a member of technical staff at Neonyoyo (now HP). He has served as program co-chair for PAISI (2006~2008), and publication chair for PAKDD 2006. Kuiyu has published over 50 papers, including in top journals (IEEE TPAMI, TKDE) and conferences (SIGIR, IJCAI, ICDE, ICDM, SDM), and is also recipient of 2 international best paper awards. Since 2005, he has been leading the ROSE (Review and Opinion Search Engine) project, which aggregates online Chinese sentiments. Kuiyu consults regularly for IT companies in China, Singapore, and Malaysia. He received his Ph.D. degree from the University of Texas at Austin, M.S. degree from the University of Hawaii at Manoa, and B.S. degree from National Taiwan University, all in electrical and computer engineering.