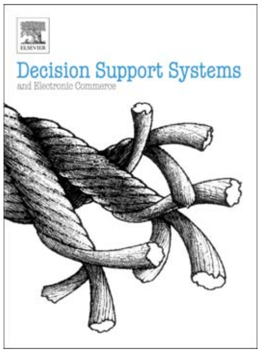

Sentiment Analysis in Financial Texts

Samuel W.K. Chan, Mickey W.C. Chong

PII: S0167-9236(16)30182-8
DOI: doi:10.1016/j.dss.2016.10.006
Reference: DECSUP 12783

To appear in: *Decision Support Systems*

Received date: 23 September 2015
Revised date: 20 September 2016
Accepted date: 31 October 2016

Sentiment Analysis in Financial Texts

Samuel W.K. Chan and Mickey W.C. Chong
Department of Decision Sciences
The Chinese University of Hong Kong
Shatin, Hong Kong
{swkchan, mickey_chong}@cuhk.edu.hk

Abstract

The growth of financial texts in the wake of big data has challenged most organizations and brought escalating demands for analysis tools. In general, text streams are more challenging to handle than numeric data streams. Text streams are unstructured by nature, but they represent collective expressions that are of value in any financial decision. It can be both daunting and necessary to make sense of unstructured textual data. In this study, we address key questions related to the explosion of interest in how to extract insight from unstructured data and how to determine if such insight provides any hints concerning the trends of financial markets. A sentiment analysis engine (SAE) is proposed which takes advantage of linguistic analyses based on grammars. This engine extends sentiment analysis not only at the word token level, but also at the phrase level within each sentence. An assessment heuristic is applied to extract the collective expressions shown in the texts. Also, three evaluations are presented to assess the performance of the engine. First, several standard parsing evaluation metrics are applied on two treebanks. Second, a benchmark evaluation using a dataset of English movie review is conducted. Results show our SAE outperforms the traditional bag of words approach. Third, a financial text stream with twelve million words that aligns with a stock market index is examined. The evaluation results and their statistical significance provide strong evidence of a long persistence in the mood time series generated by the engine. In addition, our approach establishes grounds for belief that the sentiments expressed through text streams are helpful for analyzing the trends in a stock market index, although such sentiments and market indices are normally considered to be completely uncorrelated.

Keywords: Text analysis; financial time series; decision support systems.

I. Introduction

Dealing with the deluge of data being rendered through networks of people or devices has become increasingly important for business intelligence (Chen et al., 2012). One of the notable aspects of this big data boom is the enormous breadth of the interactions between participants that can be documented. As a result of this trend, most of the data are increasingly unstructured, with a significant portion of the data stream in textual format. The forms of such data range from e-mail communications and tweets to corporate reports and daily news announcements. As this stream of data continues to expand rapidly, it grows increasingly important to develop techniques for skimming through countless pages of digitized texts and picking out the useful information that is hidden in plain sight. For professionals in the financial markets, the information boom is especially challenging. There is no shortage of news organizations and commentators who offer information or opinion about the markets through traditional websites, posts on Twitter or other social media outlets. Under the efficient market hypothesis, the efficiency of markets relies on the delivery of market information to the investors in a timely and correct manner. However, as the amounts of financial media output and market data expand at a rapid pace, perfectly informed and rational decisions are commonly unattainable, due to the cognitive limitations of the investors' minds and the finite amounts of time they have to make decisions. Under these conditions, it is not viable for any investor to comprehend the preferences of all the relevant financial text authors before making investment decisions.

As the structure of a language affects the ways in which its respective speakers conceptualize their world, one of the most important strategic trends for dealing with the current text boom is the application of text analysis (Groth & Muntermann, 2011; Cecchini et al., 2010) and related techniques for carrying out sentiment analysis of financial texts. Such techniques can reveal not only the latest trends in the public mood as reflected in the media, but also provide clues for analyzing possible ramifications and reducing the risks of conducting transactions in chaotic financial markets (Schumaker & Chen, 2009; Bollen et al., 2011; Chen et al., 2011). The application of text analysis is crucial for professionals in both business intelligence and financial markets, as they seek to build an advanced analysis capability for dealing with the current big data boom. However, the vast majority of decision support system managers have so far shown little interest in applying text analysis as a serious alternative to the canonical decision models.

Unstructured textual data have posed numerous challenges to analysts. The data are certainly not random sequences of characters, and in any case they cannot be easily normalized and readied for literal analysis. It is hard to capture the word order and its meaning. Some text analysis adopt the word-based approach, or bag-of-words (BOW) model, which permits a systematic coding of the meanings in texts by defining a vector of standardized scores for a set of semantic categories. These vectors then provide hints toward detecting over- and under-emphasis in relation to the trend of ideas found in the text. Obviously, the drawback of the BOW approach is that the text representation itself completely disregards grammar and even word order, and focuses instead on multiplicity. It is a well-known fact that most languages rely heavily, if not totally, upon word order in their grammatical systems. It seems counterintuitive to expect that meaning can remain intact when the word order is discarded.

$$\textit{Steve loves } (+) \textit{ iPhone and hates } (-) \textit{ Galaxy} \qquad (1)$$

In (1), the number of positive and negative terms is the same, but the readers would not feel that the author is expressing a neutral sentiment. In analyzing financial texts, the word-based approach of simply counting the number of positive or negative words has several limitations. First, the authors of financial texts are commonly circumspect in their use of negative language (Loughran & McDonald, 2011). They lean toward rewriting all negative content and using positive words. For example, the phrase "*did not benefit*" appears more often than the word "*lose*." Second, it is common to find that complicated sentences are used to express negative news. This writing tactic increases the cognitive load on the readers and dilutes the effect of the negative news. These limitations can be eased by capturing the sentences' shallow grammatical structure, which at least groups words that go together as phrases.

The two ends of the spectrum in text analysis are the BOW approach and the pure linguistic analysis method. In this study, we consider a trade-off between quality and complexity by providing a novel approach to sentence parsing. Then we take advantage of existing sentiment lexicons to build up a shallow but efficient sentiment analysis engine (or SAE). Different from other approaches, we consider both shapes of subtrees in parse trees and part-of-speech (POS) syntactic tags in our sentence parsing. As a result, a phrase with high grammatical complexity will be differentiated from a simple phrase and thus generates a more accurate parse tree. This study explores a novel set of context features, collectively called tree topological features (TTFs), for use in quantifying subtree configurations.

The rest of this study is organized as follows. First, the related work in sentiment analysis and its influence in finance are described in Section 2. In this research project, we make use of a machine learning technique to devise a parser that relies on various heterogeneous context features, namely POS tags, their collocation information-theoretic measures and a set of TTFs. The parser is then capable of indicating shallow grammatical structures and patterns. Section 3 presents the architecture of this parser. A parsing technique for resolving the optimal parse tree is explained in detail. Having determined the parse trees of the sentences used in financial texts, the simplest way to bring sense to our sentiment analysis is to introduce a taxonomy involving different sense categories. These categories gather words with shared meanings at different levels of abstraction. In Section 4, we introduce a heuristic technique that assesses the polarities of the parsed sentences. This technique provides a means to quantify both the cognitive and the prioritizing features of the sentences in financial texts. This engine has already been implemented using the Java language. To demonstrate its capability, the engine is deployed for parsing English and Chinese sentences using the English Penn Treebank (Marcus et al., 1993) and the Chinese Penn Treebank (Palmer et al., 2005). A detailed evaluation is provided in Section 5. In this section, in addition to a benchmark evaluation using a dataset of English movie review, we consider that financial text streams are more than just collections of isolated documents, because they exhibit the traits of collective intentions and actions. These text streams contain prominent hints for investors, as they convey numerous expressions of sentiment that are derived from the wider public. These texts are therefore helpful in evaluating our SAE. We explore whether there are any hidden sentiment patterns uncovered by our engine in relation to market indices. A collection of daily publications and blogs with twelve millions of words are processed in the engine. Statistical tests, including the Hurst exponents and Granger causality tests, are conducted to test the significance of our findings. Further directions for this research are also indicated in Section 5, followed by a conclusion.

II. Related Work

Financial texts have become more readily available due to the proliferation of postings to the Internet and the ever-increasing demands for market transparency. These trends have given rise to financial text analysis. The idea of applying textual analysis to the financial markets is not completely new and the impact of sentiment analysis on financial markets is well established. A survey by Klein and Prestbo (1974) shows how a pessimistic financial news report can affect the markets, and this study firmly supports the suggestion that news reports and markets influence each other. Ederington and Lee (1993) conclude that financial texts, particularly press releases, can shed light upon intra-market volatility. Engle and Ng (1993) suggest the notion of news impact curve, which provides a device to explain market returns using news. Wuthrich et al. (1998) analyze news articles from five popular financial websites and develop an online computational linguistics system for predicting stock prices. Melvin and Yin (2000) also suggest that readers usually pay more attention to the financial news headlines on the fly. The impacts of the headlines on financial returns cannot be ignored. Poon and Granger (2005) describe how a combination of stocks and options can be used to predict volatilities. They conclude that the best and most elaborate quantitative models fail to rival predictions based on implied volatilities. These authors note that the question of whether forecasting can be enhanced by using exogenous variables such as news reports is potentially important for future research. These ideas are extended by Chan (2003), who study the profitability of different types of portfolios. Portfolios with stocks featured in news releases outperform others over the same period, and these featured stocks have significantly high momentum returns. The exogenous information supplied by the news reports improves stock returns. Loughran and McDonald (2013) suggest the percentages of uncertain, weak modal and negative words are powerful variables in explaining levels of underpricing in most initial public offerings (IPOs) in stock markets. Supported by computational linguistics, Antweiler and Frank (2004) trace more than 1.5 million messages from *Yahoo! Finance*, and find that stock messages help predict market volatility. Tetlock (2007) shows that the number of negative words (as defined by the Harvard IV4 Dictionary) in the “Abreast of the Market” column in the *Wall Street Journal* can help to predict a company’s cash flow. The presence of pessimism in this column predicts negative returns (reversals) the next day, but this predictability disappears within a week. Stock prices usually under-react to the underlying negative information supplied by such news articles, and it takes roughly one day for negative news to affect the market. Baker and Wurgler (2006) also present evidence that investor sentiment has significant effects on stock prices. Kothari et al. (2009) conclude that adverse news about a firm is linked with its stock price volatility. Rather than using the traditional Harvard Dictionary to uncover negative information in texts, Loughran and McDonald (2011) develop an alternative negative word list that better reflects the tone of financial texts. They also count on a common-term-weighing scheme to reduce the noise introduced by misclassification in financial texts. Shiller (2000) argues that the news media play an important role in market movements. Investors tend to follow printed words, even though most financial writing is pure hype. It is one of the main reasons in creating the asset bubbles. Garcia (2013) revisits the suggestions made by Shiller (2000) by studying financial market news from the *New York Times* over the 1905 to 2005 period, and concludes that the link between media content and stock market returns is indeed concentrated in times of hardship.

Sentiment analysis draws its academic building blocks from text processing and computational linguistics. Liu (2012) observes three main types of sentiment analysis, based on their levels of granularity. First, the document-level approach tries to produce an ultimate verdict based on the entire content (Pang et al., 2002; Turney, 2002). This approach is also known as document-level classification. Second, the sentence- or phrase-level approach aims to determine the polarity of each individual sentence. Shaikh et al. (2007) integrate a semantic parsing technique to assess textual sentiment. Wilson et al. (2005) propose a phrase-level sentiment analysis that identifies the contextual polarity of a larger set of sentiment expressions. Wiebe and Mihalcea (2006) propose a machine learning method to identify the subjectivity of phrases. These authors consider subjectivity to be a property that can be determined from word sense. Third, fine-grained sentiment analysis moves from a document basis to an individual or aspect basis. The main rationale for this approach is that a negative review of a document does not necessarily imply that the reviewer is pessimistic toward all aspects found in the document (Wei & Gulla, 2010). Ensemble machine learning techniques are also applied in the sentiment analysis (Wang et al., 2014; Fersini et al., 2014). This approach particularly concentrates on phrases rather than words, given that the fundamental units of expression are phrases, and thus they are the most meaningful chunks of text. A typical application of aspect-based sentiment analysis is the assessment of product reviews (Hu & Liu, 2004; Lu et al., 2009). The studies conducted on sentiment analysis as described above have not specifically targeted financial texts, and neither have they provided any hints concerning the analysis of financial markets.

As financial texts have an undisputed role in affecting the market (Schumaker et al., 2012; Mitra & Mitra, 2011), there is a growing demand for incorporating more linguistic knowledge into the sentiment analysis of financial news. In this study, our sentiment analysis of finance data take advantage of linguistic analysis based on grammar, which extends the assessment process not only at the token level, but also at the phrase level within each sentence. This inclusiveness is possible because such linguistic analysis identifies various phrases in relation to the recursive structure of parse trees. Along with demonstrating the functionality of our mathematical economic models, we show that the patterns of sentiment uncovered by our engine help to reveal the conceptual drifts in financial markets.

III. Sentiment Analysis Engine

Our SAE first builds up a parse tree of an incoming sentence using a bottom-up derivation strategy. The engine relies on chunk-based parser that starts from a string of word tokens with a set of context features. An ensemble machine learning technique is adopted for building up two classifiers that are trained from the context features. The first classifier is to predict the right chunking point that lies between two adjacent phrases. The second classifier is to figure out the appropriate syntactic structure for the chunks, such as noun phrases. In the following section, we first provide a brief review on the basic architecture of the SAE. Interested readers can refer to the literature for more detailed discussions of the parser (Chan et al., 2011).

3.1 Basic Architecture

The SAE is divided into five major modules, namely the (i) feature extraction, (ii) sentence chunker, (iii) phrase recognizer, (iv) parse tree resolution and (v) sentiment assessment modules, as shown in Figure 1. For every input sentence in a financial text, sets of features are extracted in the extraction module. These features include the words of the sentence, their POS sequences, their collocation information-theoretic measures and a set of TTFs. The input features sequence is first fed into the chunker, which tries to locate the boundaries and the spans of the phrases or chunks. The phrase recognizer predicts the phrasal structure, which is called the non-terminal syntactic class (SC) tag of the identified chunks. Both the phrases and SC tags acquire the knowledge encoded in treebanks and take the advantage of a machine learning technique for the chunking and prediction.

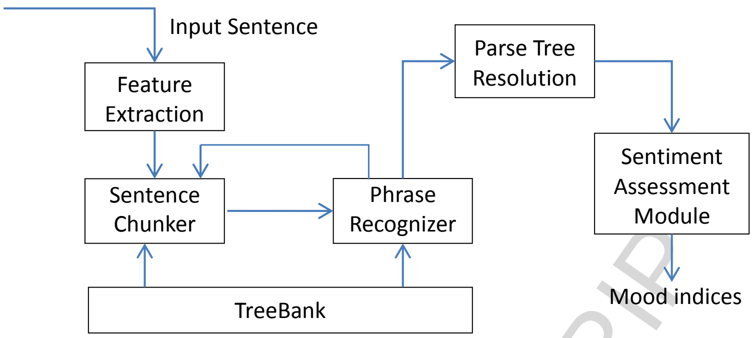

Figure 1: Architecture of the SAE

To identify the recursive phrasal structure, the *i*-th level SC tag sequence is then fed back to the chunker for processing at the (*i*+1)th level. This iteration continues until a complete parse tree is constructed. Inside the sentence chunker, a novel approach that aims at identifying chunk boundaries is applied. First, let us define an in-between point v_n as the point between two consecutive tags, u_n and u_{n+1}, in a POS or SC tag sequence. If two consecutive tags constitute a phrase in its higher level, the in-between point v_n is defined as a merging point and is marked with "+". For example, in the input sentence "*US stocks rebound from weekly drop*", we have the word tokens, "US", "stocks", "rebound", "from", "weekly", "drop" with their POS tags "NNP", "NNS", "VBN", "IN", "JJ", "NN" respectively as shown in the last two rows of Table 1.

L4	NP										
L3	NP			/	VP						
L2	NP			/	VBN	+	PP				
L1	NP			/	VBN	/	IN	+	NP		
POS	NNP	+	NNS	/	VBN	/	IN	/	JJ	+	NN
Word	US		stocks		rebound		from		weekly		drop

Table 1: Propagation of syntactic class (SC) from the POS level to the L4 level during the iterative process. All the shaded cells are in-between points since there are two consecutive POS or SC tags in the sequence. Cells marked with "+" and "/" are the merging and chunking points respectively.

Levels 1-4 (L1-L4) demonstrate the SC tag sequences. The explanation of selected POS and SC tags of the English Penn Treebank can be found in Table 2.

Type	Tag	Description	Type	Tag	Description
SC	ADVP	Adverb phrase	POS	MD	Modal auxiliary
SC	NP	Noun phrase	POS	NN	Noun, singular or mass
SC	PP	Prepositional phrase	POS	NNP	Proper noun, singular
SC	S	Root	POS	NNS	Noun, Plural
SC	VP	Verb phrase	POS	RB	Adverb
POS	DT	Determiner	POS	VB	Verb, base form
POS	IN	Preposition	POS	VBN	Verb, past participle
POS	JJ	Adjective			

Table 2: Selected POS and SC tags, with description, of the English Penn Treebank

At the POS level, the point in-between the consecutive tags NNP and NNS is a merging point, as NNP and NNS are more likely to form a NP phrase in its higher level L1. Similarly, the point in-between VBN and IN at the POS level is a chunking point and indicated with “/”, as VBN and IN are not siblings. Both VBN and IN have their own parents in its higher level L1. The main objective of our chunker is to locate all possible chunks in each level by classifying the in-between points as either merging or chunking points. On the other hand, our phrase recognizer tries to predict the SC tags from the chunks propagated from its subordinate level. For example, at POS level, the chunk <NNP,NNS> is predicted to be an NP, <NNP,NNS> → NP, <VBN> → VBN, <IN> → IN and <JJ,NN> → NP by the recognizer. By activating the chunker and the recognizer iteratively, the output strings of levels 1 to 4 chunking are determined as shown in Table 1. Figure 2 demonstrates the parse tree that can be directly reconstructed from Table 1.

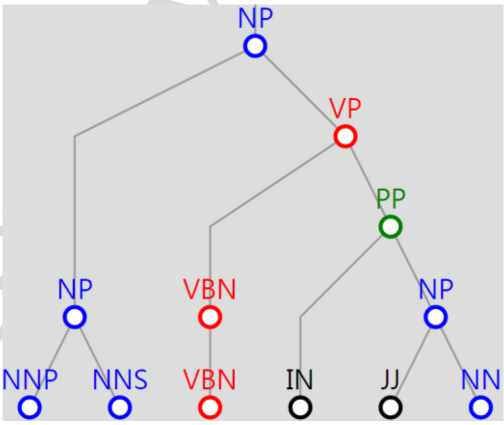

Figure 2: Parse tree of the sentence “*US stocks rebound from weekly drop*”

Table 3 summarizes the basic rationale of the chunker and the phrase recognizer. In our training, a feature vector is constructed for each training case, and the corresponding target attribute with one of the binary values (i.e., the merging vs. the non-merging point) is also provided. The sets of features are further explained in Section 3.2. For instance, at the POS level of Table 1, the input tag sequence *TS* at step 3.0 in Table 3 is <NNP, NNS, VBN, IN, JJ, NN>. Armed with the context features as described in Section 3.2, our sentence chunker differentiates the in-between points between each pair of input tags as either chunking or merging points shown in step 3.1.1. As a result, the points in-between (US/NNP, stocks/NNS) as well as (weekly/JJ, drop/NN) are merging points while the rest are all chunking ones. In step 3.1.2, our phrase recognizer predicts the resultant SC of the chunks. A new tag sequence *TS* <NP, VBN, IN, NP> is being updated and propagated to its upper level, L1, as shown in Table 1. The iteration is repeated until the tag sequence *TS* contains only one element, i.e., it reaches the top level L4.

```
1.0 prepare training data from the treebank based on a set of context features
2.0 train the chunker and phrase recognizer using an ensemble machine learning technique
3.0{ for any input tag sequence TS do
    3.1{ while TS contains more than one element
        3.1.1 differentiate all possible + or / in TS by the chunker with uncertainty
        3.1.2 predict the SC of each identified chunk, with uncertainty, in the phrase
              recognizer
        3.1.3 update TS with the new SC sequence for next level of parse tree generation
    }3.1 end while
    3.2 calculate the overall certainty of all possible parse trees of TS
}3.0 end for
4.0 feed all possible parse trees with their certainty to the parse tree resolution module
```

Table 3: Basic rationale of the chunker and phrase recognizer modules. Numbers in bold represent the steps of execution.

3.2 Context Features

In this study, we use three broad categories of context features to capture the likelihood that one or more chunks will turn into a new phrase. The first type of context features is the word order that takes a pivotal role in understanding sentences. It is definitely the most fundamental syntactic device in all languages. In our parser, we capture the word order feature using a sliding window with size equal to six. In other words, three SC tags preceding the in-between point v_n as well as the three tags following the v_n are used to represent the word order context feature. The second type of context features is the measure of association, namely the pointwise mutual information (PMI) and the likelihood ratio (LR) of the SC tags, which reflect the likelihood of the chunk collocation. Various adjacent POS/SC fragments in the neighborhood of u_n and u_{n+1} are defined.

u_{n-2}	u_{n-1}	u_n	u_{n+1}	u_{n+2}	u_{n+3}	Association features
	u_{n-1}	u_n				d_1: $\zeta(u_{n-1}, u_n)$
		u_n	u_{n+1}			d_2: $\zeta(u_n, u_{n+1})$
			u_{n+1}	u_{n+2}		d_3: $\zeta(u_{n+1}, u_{n+2})$
u_{n-2}	u_{n-1}	u_n				d_4: $\zeta(u_{n-2}u_{n-1}, u_n)$
	u_{n-1}	u_n	u_{n+1}			d_5: $\zeta(u_{n-1}u_n, u_{n+1})$
		u_n	u_{n+1}	u_{n+2}		d_6: $\zeta(u_n, u_{n+1}u_{n+2})$
			u_{n+1}	u_{n+2}	u_{n+3}	d_7: $\zeta(u_{n+1}, u_{n+2}u_{n+3})$

Table 4: Measure of association in various adjacent SC chunks, where the in-between point v_n is between u_n and $u_{n+1.}$ ζ denotes the association measures using pointwise mutual information (PMI) or the likelihood ratio (LR) of the chunks.

Table 4 summarizes the association measures computed for training the classifier. Taking the in-between point v_n between `IN` and `JJ` in Table 1 as an example, *d*6 represents the PMI between `IN` and (`JJ NN`), i.e., *PMI*(`IN`, `JJ+NN`). While the pointwise mutual information compares the probability of observing two different chunks together with the probability of observing them by chance, the LR is a formalization of independence, which provides another index for measuring the degree of association between two different chunks. Again, taking the in-between point v_n between `IN` and `JJ` at the POS level in Table 1 as an example, the degree of association using LR in *d*6 represents the *LR*(`IN`, `JJ+NN`). Two alternative hypotheses are examined as shown in (2).

$$H_0: P(\texttt{JJ+NN} \mid \texttt{IN}) = p = P(\neg\texttt{JJ+NN} \mid \texttt{IN}) \qquad (2)$$

$$H_1: P(\texttt{JJ+NN} \mid \texttt{IN}) = p_1 \neq p_2 = P(\neg\texttt{JJ+NN} \mid \texttt{IN})$$

Other than the word order and its association, researchers in text analysis have reported that the number of tokens in a phrase, or called "weight" in a chunk, has impacts on the structures of sentences (Rosenbach, 2005). Complex chunks with heavy weight tend to appear at the end of a sentence (Wasow, 1997). In this study, we define a set of TTFs that describe the shapes of subtrees quantitatively. These features include the following:

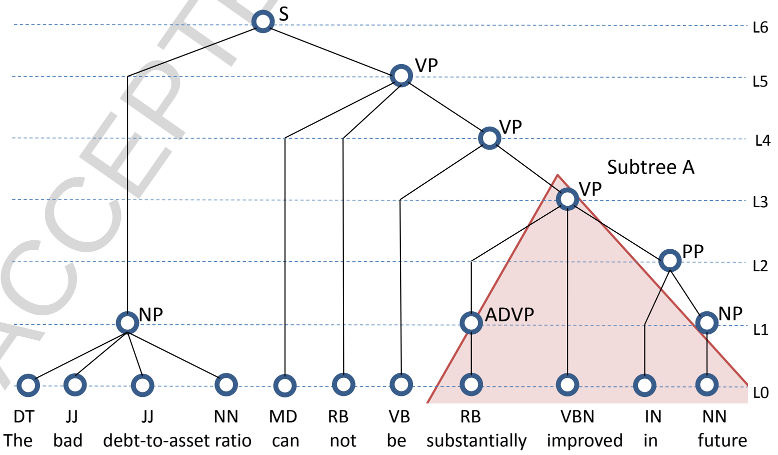

Figure 3: Topological features of a subtree in sentence `S`

- *Span Ratio (SR):* The *SR* is defined as the total number of tokens spanned under the target SC tag, divided by the length of the sentence. In Figure 3, the *SR* for the target SC tag `VP` at subtree *A* is 4/11. This ratio reflects the weight of the SC tag and demonstrates how far the target SC tag is from the root `S`.
- *Aspect Ratio (AR):* The aspect ratio of a subtree is the total number of SC tags divided by the number of terminal nodes at L0. In Figure 3, the *AR* for the `VP` at subtree *A* is 4/4.

- *Node Coordinates (NCs): NCs* involve two parameters, namely the level of focus (*LF*) and the relative position (*RP*) of the target subtree. Whereas *LF* indicates the current level of focus, the relative position points out the order of the target tag in that level.
- *Skewness Measure (SM): SM* provides a likelihood function that measures the degree of right branching of a subtree. It takes into account of the path lengths connecting a target SC tag and its terminal nodes. The path length between two nodes is the number of edges traversed between the nodes. For example, in the subtree A in Figure 3, the path length between the top `VP` and `NN` at terminal level is equal to 3. For a subtree with *n* terminal nodes, there are *n* paths. A pivot that provides an axis of vertical flipping of a subtree is defined as the [*n*/2]th terminal node if *n* is odd, and as between the [*n*/2]th and [(*n*+1)/2]th terminal nodes if *n* is even, where [ ] is a ceiling function. For example, the pivot of the subtree A in Figure 3 lies between `VBN` and `IN`. We then define *SM* as

$$SM = \frac{1}{\sum_{\rho_i > 0} \rho_i} \left(\frac{\sum_{i=1}^{n} \rho_i \left(x_i - \bar{x}\right)^3}{\sigma^3} \right) \tag{3}$$

where x_i is the *i*-th path length connecting the target SC tag and the *i*-th terminal node, $\bar{x}$ and σ are the mean and standard deviation of all the lengths at the relevant level, and ρ_i is a moment distance of the *i*-th terminal node to the pivot. For example, in subtree A, x_i are equal to 2, 1, 2, 3 while ρ_i are 2, 1, -1, -2 for *i*=1..4. These figures suggest the subtree is slightly right branching.

3.3 Classifications Using an Ensemble Machine Learning Technique

Even though the claim that text always repeats itself is too strong, we must accept that there are many similar phrases/sentence patterns that appear in texts. Although we try to capture the regularity indicated by the aforementioned supra-syntactic clues (which certainly provide context sensitivity), we believe that regularity can be easily spotted using the latest inductive learning techniques. In the last few decades, learning algorithms such as decision trees (DTs) and support vector machines (SVMs) have performed excellently, reducing the potentially negative effects from unrealistic assumptions of independence among features. A DT consists of nodes in which attributes are tested. The learning adopts a top-down strategy that divides the search area into rectangular regions. This method guarantees that a simple tree, but not necessarily the simplest tree, will be found. Even the best algorithm may perform poorly in some scenarios and algorithms that have poor average performances often perform well in solving a few problems or metrics. The SVM provides a different but complementary technique that has been proven effective in handling the large number of features encountered in our training tuples. The SVM uses a nonlinear mapping method to transform the original training data into a higher dimension (Joachims, 2002). Within this new dimension, the SVM searches for the linear optimal separating hyperplane that is a decision boundary for separating the tuples of one class from another. In our application of decision tree algorithm, we take advantage of the latest version of C4.5 (Quinlan, 1994) with its default pruning ratio. We adopt the boosting with 10 trials. As observed in the dataset, each discrete attribute may have more than 10 possible values, particularly in the POS and SC tag. We apply the discrete value subsets option (`-s`) to eliminate the undesirable side-effect of fragmentation of data during the construction of the decision tree. For the SVM model, we apply the *C*-support vector classification (*C*-SVC) with a typical radial basis function (RBF) kernel with default parameters. The classifier is under a standard Library for Support Vector Machines (Chang & Lin, 2011). During training, we set the tolerance of the termination criterion to 0.001. Both the DT and SVM algorithms apply their diverse types of discernment to the tasks of chunking and phrase prediction. However, none of these algorithms can achieve a perfect prediction when applied in isolation. The algorithms may provide two feasible predictions, or sometimes vague or even contradicting predictions. In our study, an ensemble technique is applied in combining the algorithms in a meta-level classifier and building up the parser. First, we introduce three heterogeneous and mutually independent context features, namely the word order, the association measures and the tree topological features. Instead of training all three types of context features to form a single gigantic classifier, we produce six different base-level classifiers from the two learning algorithms, in the hope that they will produce a highly accurate prediction. At the same time, we apply the meta-decision tree (MDT) that learns to combine predictions from the six different base-level classifiers (Ženko et al., 2001). Basically, the rationale of the MDT is quite similar to that of the other base-level decision trees, such as CART or CHAID, except that each leaf node of the MDT indicates the base-level classifier that will be adopted for prediction. In other words, the target class feature assigned to the leaf node of the MDT determines which base-level classifier should be triggered. The input features of the MDT are the verdicts, their confidences and the set of meta-level attributes deduced by the relevant base-level classifiers. To assess the certainty and confidence of the predictions generated from each base-level classifier, the MDT defines an entropy measure and a maximum probability for the target class probability distribution of the base-level classifier. In other words, when a base-level classifier exhibits a low maximum probability and a high entropy, this pattern signals that the classifier is quite skeptical about its prediction of the target class value. In addition, a weight function, `weight`(*x*, *CL*), is defined as the fraction of the training samples used by the classifier *CL* to estimate the target class distribution for sample *x*. This function quantifies how reliable the predicted class probability distribution is. Intuitively, the weight corresponds to the number of training samples used to estimate the probability distribution: the higher the weight, the more reliable the estimate. Interested readers can refer to the literature for more detailed discussions (e.g., Todorovski & Džeroski, 2003).

In sum, this technique aims to effectively solve combinatorial optimization problems. The discriminative nature of the meta-level classifier that emerges from our feature sets allows our chunker to identify the optimal chunks based on the supra-syntactic clues. In doing this, we safeguard chunking accuracy while providing sufficient tolerance for errors that may arise from POS tagging and even from basic word tokenization, which are fundamental and often non-trivial aspects of various languages, including Chinese.

3.4 Parse Tree Resolution

As shown in Table 3, for any input sentence there is more than one possible parse tree generated by the chunker and the phrase recognizer. Each of the choices made by the chunker and recognizer is associated with an uncertainty. To establish the final parse tree, a chart parsing representation is used to resolve the local ambiguities or the incomplete structures found (Billot & Lang, 1989). The basic rationale of this representation is that it always keeps track of partial derivations that are suggested by the chunker and the recognizer, and it never throws away information concerning any possible parses until the final resolution has been made. The major data structure in the representation is a chart that provides a flexible framework to store intermediate parse results, and thus allows for an efficient retrieval regardless of any parsing strategies. Although it is only a data representation, this chart enables different types of parsing algorithms to be developed and tested. The well-known CYK and Earley parsers are two typical examples of the uses for such chart representation. Statistical sentence parsing usually begins with an agenda *E* and an empty chart *C*. The parsing starts by first scanning all words in the input and placing the parts of speech (with uncertainty in terms of their probability) into the cells associated with span one. Starting from the word level in *C*, the parsing continues by subsequent applications of the grammar production rules to fill the remaining cells, from the smallest to the largest span. As most grammars are ambiguous, there can be more than one possible alternative for non-terminals. The chart representation is therefore usually extended by adding back-pointers to the items in the charts. The back-pointers reveal the constituent children. As a result, a single parse tree can be read off from the chart by starting from the root at the apex of the chart and tracing the back-pointers.

As with other parsing algorithms, our parsing model allows multiple derivations to "sprout" and propagate more than one possible phrase structure up the root by using a chart representation. Our chart entry consists of a 5-tuple (A, CF, α, β, k), where A is the non-terminal SC of an identified chunk; CF represents the confidence (or the certainty level) of the SC as predicted by the meta-classifiers in the chunker and the phrase recognizer; the back-pointer α indicates a list of children that build up the current node A, and the location index of the children is recorded in the list β and in the current location index k. Each cell entry maintains a list from the back-pointer α that enumerates the different ways in which the A can be built in that cell. This list allows us to trace the sources of the list item subordinates. Each chart entry can house more than one tuple. The tuple representation provides a means to represent the resulting parse forest, and it renders sufficient information to reconstruct all possible parse trees that sprout from its terminal level. Unlike other researchers, we impose no restrictions on the length of the chunks or on the number of children in the back-pointer α. In other words, our approach requires no binarization that may hinder the parsing by inflating the number of grammar productions through endless but unnecessary derivations from the terminal level to its root. Our chart parsing technique is sketched in Table 5.

```
$C \leftarrow \varnothing$; $E \leftarrow \varnothing$
for all items $T$ segmented from the chunker and predicted by the phrase recognizer
    do
    $C \leftarrow C \cup T$; $E \leftarrow E \cup T$
end for
while $E \neq \varnothing$
    remove an item $T$ from $E$
    for all items $T'$ resulting from the chunker and phrase recognizer with
        antecedents $T$ and items from $C$ do
         if $T' \notin C$ then
            $C \leftarrow C \cup T'$; $E \leftarrow E \cup T'$
         end if
    end for
end while
select the root of the optimal parse tree with a higher certainty factor at the root node
trace all branches of the optimal parse tree using the back-pointers
```

Table 5: Chart parsing technique in resolving the optimal parse tree

First, at each level of parsing in the bottom-up trail, the MDT predicts all possible parses with certainty. The parses are then subjected to a set of constraints before filling the possible phrases in the chart, in the hope that the number of derivations will not increase exponentially. The constraints include (i) keeping a single entry per phrase structure in each cell of the parsing chart, (ii) identifying whether a phrase structure is a dead-end that cannot subsequently combine with other phrase structures into a valid parse and (iii) reducing the number of phrase structures in each cell by imposing a predefined threshold to avoid wasting any subsequent processing on that phrase. In summary, even though the chart and the back-pointers can accommodate a compact representation of a parse forest, our strategy involves propagating a subset of possible derivations, each with high certainty, up the root. First, at each level we make use of various features in segmenting the right chunks of the sentence. Second, we check whether a reasonable phrase is selected at each level of parsing in the bottom-up trail, and then select the most favored tree at the apex of the parsing chart.

IV. Sentiment Assessment Heuristics

Sentences and their syntactic structures are not important in and of themselves. They are important because they are shaped by an author's experiences and point to his or her expectations or predictions. Any text analysis system that relies solely on the orthographic structures of phrases, without any means to determine sense, will be rather shallow. The simplest way to bring sense into our sentiment analysis is to introduce a taxonomy with different sense categories that bring together words with shared meanings at different levels of abstraction. In English, we take advantage of the SentiWordNet 3.0, which is an enhanced lexical resource that is explicitly devised for supporting sentiment classification (Baccianella et al., 2010). This lexical resource results from the automatic annotation of all the synsets of the well-known WordNet, according to the notions of "positivity," "negativity," and "neutrality." Each synset *S* is associated with one of three numerical scores: POS(*s*), NEG(*s*) and NEU(*s*). These scores are called valences, and they indicate how positive, negative or neutral the terms contained in the synset are. For the Chinese language, even though there are a wide range of Chinese taxonomies in existence, few of them are widely accessible. Among the cognitive taxonomies that are available electronically, the Chinese Linguistic Inquiry and Word Count Dictionary (LIWC) is perhaps the most applicable. The LIWC contains more than 6,800 words in 30 linguistic and 42 psychological categories (Huang et al., 2012), and it provides a sufficient number of categories to analyze the complexity of human cognitive behavior.

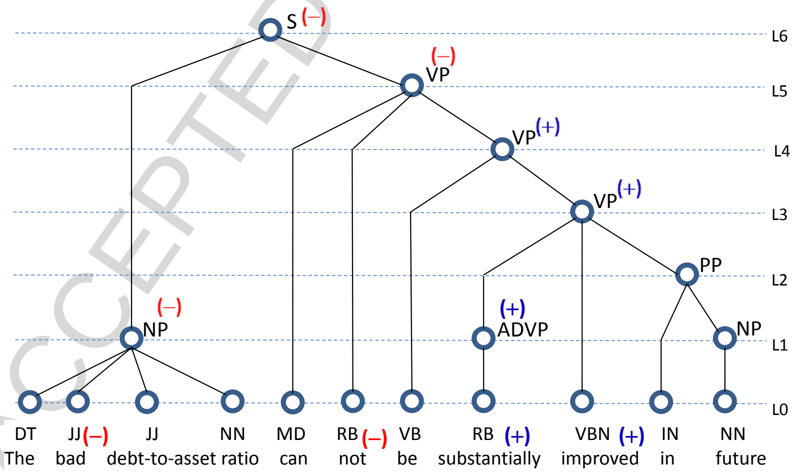

Figure 4: Parse tree of the sentence *The bad debt-to-asset ratio cannot be substantially improved in future*. The words, *substantially* and *improve* are positive, but *bad* and *not* are negative in valence. The valences of the words are radiated up to the root *S*.

Obviously, it would be highly misleading to determine the polarity of a sentence by simply counting the number of positive or negative words. The polarity of a sentence resides not only in the words and taxonomy used, but also in how the words relate to others. Human language enables authors to express complicated concepts in the form of a highly recursive structure known as grammar. It is important to fully understand this recursive structure to determine the positive or negative contextual polarity of a complete sentence. As shown in the previous section, we first unveil the structure of the sentences and then output the parse tree. Figure 4 shows the parse tree of the sentence "*The bad* *debt-to-asset ratio cannot be substantially improved in future*.”

The effects of terms on the polarity of the whole sentence are calculated when the valences of the lexical items are propagated from the POS level (L0) to the top root (L6). Table 6 gives our sentiment assessment heuristics.

```
Input: A parse tree T of sentence S and head words w of every phrasal nodes in T
retrieve the valence of the words at the POS terminal level, L0;
$L \leftarrow 1$ ;
while not (root of T) do
    for every possible phrase in level L
        identify the head words w at the level
        calculate the effects on the polarity of w by its siblings
        radiate the consequent polarity of w up to its parent
    end for
    $L \leftarrow L + 1$;
end while
Output: Polarity of the whole sentence S
```

Table 6: Sentiment assessment heuristics

Words with prior valences appear in financial texts frequently, and this leads us to adopt the heuristics described above. First, we retrieve all prior valences of the words in English and Chinese sentences from the SentiWordNet and the LIWC, respectively. In text analysis, not every word is equally important in the sentence. A head word of a phrase is the word that determines the semantic category of that phrase. The other elements of the phrase, or the siblings of the heads, are used to modify the head. To capture the main gist of the sentence, it is imperative to identify the heads of every possible phrase in the sentence. For example, the head of the NP *The bad debt-to-asset ratio* is the noun *ratio*, which is modified by its siblings, *bad*/JJ and *debt-to-asset*/JJ, as shown in Figure 4. In this study, we adopt the head rules for the English and Chinese treebank grammars from Collin (1999) and from Sun and Jurafsky (2004), respectively. Second, the polarity of a sentence is resolved by using the polarities of the phrasal structures beneath the root *S*. This computation indicates the effects of each sibling term on the polarity of the head word *w* of every possible phrase, when radiating from the terminal level L0. The following tactics are considered:

(i) The more intense the feelings of the person about the subject, the higher the polarity. To achieve this measurement, our linguistic analysis detects linguistic features such as the semantic strength of the vocabulary or the use of adverbs. Adverbs are used to modify an adjective or verb as an adverbial adjunct. Some adverbs are affirmative, such as *substantially*, or they may indicate degree, such as *very* or *quite*, even though there is no shift in polarity. Buoyed by the affirmative adverb *substantially*, the adverb phrase (ADVP) in L1 of Figure 4 further confirms the positive polarity of the word *improved*/VBN as propagated from L0.

(ii) Negation is the most obvious feature that twists the polarity of a word or phrase from positive to negative, or vice versa. As shown in Figure 4, although the phrase *substantially improved in future* has a strong positive valence, the verb phrase (VP) in L5 is twisted to a strong negative by the word *cannot*.

(iii) The conjunctions in phrases most often hint at the coordinate structures (CSs), for instance, the conjunction *and* in the phrase *bad debt-to-asset ratio and poor credit rating*. In general, phrases in CSs, such as *bad debt-to-asset ratio* and *poor credit rating*, have parallel syntactic structures and comparable polarity. In our heuristics, the polarity of the phrase in the CS with maximum absolute value diffuses up to its parent level.

(iv) Additional rules for estimating the effects of the sentence-based polarity are delineated in Table 7. These rules cater to the different possible combinations of phrasal structures found in various levels of the parse tree. For example, in Figure 4, an NP with negative polarity such as *bad debt-to-asset ratio*, is modified by the negative VP, *cannot be improved*. Consequently, the polarity of the whole sentence becomes negative (rather than neutral) by simply counting the number of positive or negative valence words at the terminal level L0.

POS of head word (*H*)	Polarity of *H*	Strongest sibling (*S*)	Polarity of *S*	Resultant polarity	Example
Noun/NP	Positive	ADJ/ADJP	Positive	Positive	*good price*
Noun/NP	Positive	ADJ/ADJP	Negative	Negative	*inappropriate transaction*
Noun/NP	Negative	ADJ/ADJP	Positive	Negative	*new infringement*
Noun/NP	Negative	ADJ/ADJP	Negative	Negative	*fatal attack*
Verb/VP	Positive	ADV/ADVP	Positive	Positive	*feel good*
Verb/VP	Positive	ADV/ADVP	Negative	Negative	*scarcely rise*
Verb/VP	Negative	ADV/ADVP	Positive	Negative	*often fail*
Verb/VP	Negative	ADV/ADVP	Negative	Negative	*hurt himself badly*
Noun/NP	Positive	Verb/VP	Positive	Positive	*market blooms*
Noun/NP	Positive	Verb/VP	Negative	Negative	*selling prices drop*
Noun/NP	Negative	Verb/VP	Positive	Negative	*bankruptcy increases*
Noun/NP	Negative	Verb/VP	Negative	Negative	*corruption deteriorates*

Table 7: Additional rules to reckon the effects of sibling terms on the polarity of the head word *w* in different phrases

By and large, our sentence parsing, valence tagging and sentiment assessment heuristics provide a means to quantify both the cognitive and priority features of the sentences in financial texts. These methods need not be comprehensive; they merely need to be sufficient for providing a reasonable guess concerning the main contextual preferences indicated by the sentences' authors. As an analogy, vitriolic critics of a company can be easily spotted by skimming the sentences in their texts.

V. Experiments and Evaluation

5.1 Experiments using Two Penn Treebanks

In our first evaluation, the proposed parsing model was trained and tested using the English and Chinese Penn Treebanks (Marcus et al., 1993). A treebank is a parsed text corpus that annotates syntactic sentence structures, and such treebanks are major test beds for most linguistic theories regarding sentences. Two English and Chinese parsers were built upon the two Penn Treebanks. The English Penn Treebank has 48 POS tags and 14 SC tags, and the Chinese Penn Treebank has a tagset of 33 POS tags and 22 SC tags. The development test results, including the training and test results of the chunker and phrase recognizer using the two treebanks, are shown in Table 8.

	English Penn Treebank (v. 3.0)			Chinese Penn Treebank (v. 5.0)		
	Train. cases	Test cases	Accuracy	Train. cases	Test cases	Accuracy
Sentence Chunker	4,931,561	297,411	96.4%	2,238,387	244,110	96.9%
Phrase Recognizer	3,469,890	210,520	99.2%	1,705,486	103,352	98.9%

Table 8: Development test results in the two treebanks

The development test results highlight the performances of the chunker and the phrase recognizer. All of the node elements found in the non-terminal levels contributed to our training cases. Both modules were quite robust, as illustrated by their accuracy. For example, the English chunker had an accuracy of 96.4%, and the phrase recognition for both languages was 99% accurate. However, we must be cautious about these figures, as the development test results shown in Table 8 did not take the propagation of errors from lower levels of each syntactic tree into account. After using the chart parsing technique for resolving the optimal parse tree, as demonstrated in Table 5, our overall parsing performance of the two different parsers are reported in Table 9. The overall result was that our English parser achieved an *F*-score of 87.96%, and the Chinese parser achieved an *F*-score of 85.61%. Although the *F*-score for state-of-the-art English parsers lies between 87 and 89%, it is noteworthy that the performance of our Chinese parser was rather encouraging and worked quite well. For instance, using lexicalized parsers in the Chinese Penn Treebank, Bikel (2004) obtains an *F*-score of 81.2% in parsing

Chinese sentences.

English Parser			Chinese Parser		
Labeled Recall	Labeled Precision	*F*-score	Labeled Recall	Labeled Precision	*F*-score
88.53%	87.40%	87.96%	85.91%	85.32%	85.61%

Table 9: Parsing performance of the two parsers

5.2 Evaluation using Movie Review Dataset

Our second evaluation of the sentiment engine is based on the sentence polarity dataset v1.0 that was originally released by Pang et al (2002). The data mainly contains English movie reviews that are collected from the Rotten Tomatoes website. It has been used as a de facto benchmark for evaluating sentiment applications. The dataset includes two files, each contains 5,331 short paragraphs marked with either positive or negative polarity. In this evaluation, two English annotated word lists with polarity are applied to our sentiment classification. First, the General Inquirer (GI) was first developed as a computer-assisted approach for content analyses of textual data. The GI contains 11,788 words that mainly come from the Harvard and Lasswell dictionaries. Among 182 possible tags for each word, there are two tags “Positive” and “Negative” indicated the polarity of the words. However, only 1,915 and 2,291 words are marked with positive and negative polarity respectively. Our second annotated word list is the SentiWordNet 3.0 that are derived from WordNet 3.0 with more than 117,600 synsets. Each synset is tagged as positive, negative or neutral polarity with their ranges from 0.0 to 1.0. In other words, a synset has non-negative scores for all the three categories. Different from the GI, SentiWordNet allows both positive and negative polarity assigned to the same synset. To achieve a direct comparison of the performance of the two word lists in our binary polarity classification task using the movie sentence dataset, we derive a simple subjectivity measure of a word *w* by subtracting the score of negative polarity from the positive one, i.e., subjectivity(*w*) = positive score(*w*) – negative score(*w*).

In this experiment, we evaluate the subjectivity of a paragraph *S*(*p*) in the movie dataset by considering all subjectivity clues from its sentences. At the same time, we experiment with several models and compare their prediction accuracy. We randomly select 40% of the paragraphs in the dataset for the calibration of different model parameters. The rest are reserved for testing purpose. For this well-balanced dataset, we apply the overall classification accuracy, i.e., number of correctly predicted instances over the total number of instances, as the measure of performance in different models. Our models include:

(a) Pure random model (Model 1): This is a baseline model that classifies all the paragraphs in the dataset as positive.

(b) Bag of Words (BOW) model (Model 2): This model disregards any grammar and word order patterns. It assumes a very simple text representation and assigns an equal term weight, usually 1, for the words that could be found in the wordlists and 0 otherwise.

(c) BOW with tf-idf model (Model 3): This model is similar to the BOW model but assigns different term weight to reflect the relative importance of the words in the paragraphs. A word with large tf-idf is a high frequency term within the paragraph and low paragraph frequency of the term in the dataset. This is a standard weighting factor used in most information retrieval applications.

(d) Pattern-based approach with consideration of negation (Model 4): On top of Model 3, this model also takes into account of negation expressions that shifts the polarity of a word from positive to negative, or vice versa. The negation expressions are usually hinted by shifters, such as *not*, *no*, *never*, *yet*, *none*, *but* and *without*, to name a few. For each subjective word, a preceding window with size equal to 4 is scanned for shifters. The polarity of a subjective word will be twisted if a shifter is found.

(e) Pattern-based approach with consideration of negation and modal verbs (Model 5): Similar to the negation expressions, modal verbs are used to indicate modality ranging from possibility such as *may*, *might* to necessity such as *must*, *should*. Similar to the Model 4 as above, a preceding window with size equal to 4 is scanned for the shifters and the modal verbs.

(f) Our SAE approach (Models 6-10): We apply our sentiment analysis engine as described in this research to the dataset. There are more than 800 possible assignment rules that can be devised from the treebank as demonstrated in Table 7. Each of them reckons the effects of sibling terms on the polarity of the head word in different phrasal structures. In Model 6, we apply the top 20 assignment rules that represent the most frequent phrasal structures found in the treebank. Other phrasal structures that fall outside of these 20 assignment rules will be assumed as neutral in polarity. Similarly, in Models 7-10, we exploit the top 50, 100, 200 and 300 frequent phrasal structures that have the broad coverage of more than 65%, 82%, 90% and 93% of all phrasal structures in the treebank respectively.

Model	Model accuracy using GI as the word-list	Model accuracy using SentiWordNet as the word-list
(1) Pure random (Baseline)	0.498	0.498
(2) Bag of Words (BOW)	0.582	0.686
(3) BOW with tf-idf weighting factor	0.597	0.714
(4) Pattern-based approach w/ consideration of negation	0.623	0.741
(5) Pattern-based approach w/ consideration of negation and modal verbs	0.625	0.749
(6) Our SAE approach w/ top 20 assignment rules	0.656	0.806
(7) Our SAE approach w/ top 50 assignment rules	0.665	0.815
(8) Our SAE approach w/ top 100 assignment rules	0.679	**0.821**
(9) Our SAE approach w/ top 200 assignment rules	0.679	0.821
(10) Our SAE approach w/ top 300 assignment rules	0.679	0.820

Table 10: Experimental results, in terms of accuracy, of different models in our rule-based unsupervised sentiment classification using two different word lists.

Table 10 shows the classification accuracy in different models. Even though the overall accuracy is increased by 20% in the BOW model from the baseline in the SentiWordNet word list, this unweighted representation of text consistently underperforms in all other models, regardless of word lists. There is a slight improvement in accuracy when applying the weighting factor tf-idf in the BOW approach. The pattern-based approach, as shown in Models 4 & 5, exhibits there is an improvement over the BOW approach in the two word lists. The simple pattern-based approach with the consideration of negation increases the prediction accuracy by 3%. Surprisingly, the contribution of modal verbs into sentiment analysis is insignificant. The inclusion of modal verbs in the pattern-based approach boosts the accuracy by less than 1%. The low coverage of modal verbs in the dataset may account for the finding. In Models 6-10, accuracy gains are significant in the SentiWordNet word list. As compared with the BOW approach, there are more than 11% gains in the five models. This is because our SAE approach considers the importance of recursive and non-recursive phrasal structures and is capable to identify the subtle polarity shift when aggregating phrasal polarity scores to determine the overall sentence polarity, while it is completely ignored in most learning-based sentiment analysis approaches. As our approach has reached a plateau shown in Model 9, there is no strong evidence to include the top 300 assignment rules as in Model 10, which suffers from a diminishing return. Generally speaking, all the models using SentiWordNet word list outperform the General Inquirer counterparts. The underlying reasons may include: (i) there is a great difference between the sizes of two word lists. SentiWordNet provides a broad coverage of word polarity than the Inquirer; (ii) SentiWordNet considers different part-of-speech of a word, which has a decisive role in understanding real world texts; (iii) SentiWordNet also demonstrates the relative strength of the word polarity while the Inquirer is on an all-or-none basis. This elevation provides evidences for our SAE on enhancing the classic dictionary-based sentiment analysis approaches.

5.3 Evaluation of Sentiment Using an Online Financial Text Stream

Financial text streams are more than just collections of isolated documents, in that they exhibit the traits of collective intentions and actions. These text streams contain prominent hints for investors, as they convey many expressions of sentiment drawn from the public. Such texts were thus helpful in evaluating our sentiment analysis model. In our third evaluation, we explored whether there were any hidden patterns of sentiment that could be uncovered by comparing the results of our SAE as applied to financial text streams with the data streams related to market indices. We obtained collections of financial news that were posted between September 9, 2014 and March 24, 2015 on a Hong Kong financial website called Finet. This site is one of the most influential newswires and is an authorized newspaper agency in both English and Chinese for the region. Two major collections of texts were included: (i) dailies, which included daily financial news from various newspapers along with all scheduled/unscheduled announcements and (ii) blogs, which gave comments/advice from financial analysts. The contents of the website provided comprehensive coverage of public sentiment toward the market. Table 11 gives some basic information on the collections analyzed.

Period of the experiment	Sept. 9, 2014 – Mar. 24, 2015
Number of days in the period	196
Number of trading days in the period	133
Number of news articles in the collections	46,842
Number of sentences	866,628
Average sentence length	13.82
Total number of words in the collections	11,976,799

Table 11: Some basic figures on the collections of financial news used in the evaluation

All of the sentences were fed into our SAE, which examined the positive and negative moods appearing in each article, as shown in Figure 1. As the broad-coverage of the financial text corpus like this, there are, not surprisingly, a lot of out-of-vocabulary (OOV) words without any POS tags. In our approach, a recursive technique is employed to assign the possible part-of-speech tags to all the unknown words so as to reduce the explosive ambiguity in our sentence parsing (Chan and Chong, 2013). The daily positive and negative mood indices were then calculated by taking the sum of the respective indices found in all of the articles. In our experiment, the correlation coefficient between the daily positive and negative moods over the 133 trading days was 0.00251. In other words, the indices of these moods were two independent variables that had no direct relationship. At the same time, we defined a net daily (ND) mood index that measured the difference between the daily negative and positive moods. In addition, we took a further step in normalizing the series using the z-score standardization (Bollen et al., 2011; Eickhoff & Muntermann, 2016). This was achieved by subtracting the population means from the raw scores and then dividing them by their standard deviations. This normalization transformed all three time series with zero mean and provided a common scale for comparisons of the three mood time series. Summary statistics of the three mood time series are shown in Table 12. The three time series had rather constant means and standard deviations. The statistical figures indicated the negative mood index were slightly higher than the positive ones. These results seem to suggest that the financial texts were negative. We then adopted the Hurst exponent to test whether there was a long memory or persistence in the time series.

	Positive mood index	Negative mood index	Net daily (ND) mood index
Mean	0.84513	0.94699	0.10186
Medium	0.83064	0.93741	0.08439
Minimum	0.59150	0.61236	-0.30291
Maximum	1.12142	1.57737	0.68213
Standard deviation	0.10611	0.16460	0.19168
Hurst Exponent (H)	0.80172	0.89043	0.93228
Std. error of H	0.02437	0.07639	0.04603

Table 12: Summary statistics, including the Hurst exponent, of the three mood time series

The Hurst exponent lay between 0 and 1. This exponent quantifies the relative tendency of a time series to either regress strongly to the mean or to cluster in a particular direction (Hurst, 1951; Mandelbrot, 2004). A Hurst exponent in the range 0–0.5 indicates a time series with long-term switching between high and low values in adjacent pairs, which is sometimes called a Joseph effect. This term refers to the Old Testament story in which Egypt experienced seven years of rich harvests followed by seven years of famine. A Hurst exponent equal to 0.5 indicates a completely uncorrelated series, and a Hurst exponent in the range 0.5–1.0 indicates a time series with a long-term positive autocorrelation and strong persistence. As shown in Table 12, the Hurst exponents for the three mood time series ranged from 0.8 to 0.93, which suggested that the time series were absolutely not random stationary series. All three mood series exhibited positive long-range dependence or persistence. This phenomenon coincided with most trending behavior, such as the research in sentiment analysis or financial time series (Allen et al., 2015; Couillard & Davison, 2005). In addition, the Hurst exponents for the negative and ND mood indices seemed to indicate that the negative sentiments from articles appearing in the financial media took more days to unfold. These results suggested that bad news tended to occupy the media longer than good news. Although the Hurst exponents demonstrated that our mood indices generated from our SAE were unlikely to be random time series, we were concerned with the question of whether the daily mood indices could provide any hints concerning changes in the Hang Seng Index (HSI), a major stock market index in the region. In our last evaluation, we applied the well-known Granger causality analysis (GCA) to the three time series against the HSI. The GCA is a method for investigating whether one time series can correctly shed light on another (Granger, 1969). The analysis rests on the assumption that a variable X is said to “Granger-cause” another variable Y, if Y can be better estimated from the pasts of X and Y together than by the past of Y alone (Pierce, 1977). In other words, Granger-cause is indicated if changes in X are shown to systematically occur before changes in Y.

In our evaluation of sentiment generated by the SAE, we tested for Granger causality from the mood indices to the stock index. Our stock index time series S (which was an endogenous variable) was defined to reflect daily changes in the closing price of the stock market, i.e. its values were the daily difference between day t and day t-1, $S_t = HSI_t - HSI_{t-1}$. The mood index (an exogenous variable) was denoted by M. As suggested by Bollen et al (2011), the mood index M will not subject to any transformation or z-score normalization in order to avoid so-called “in-sample” bias. In other words, all the exogenous variables were the raw scores generated from the SAE. To test the hypothesis H_1 that M Granger-causes S, we try to identify the lagged values of M that exhibit a statistically significant correlation with S. The autoregression in H_0 was augmented by the lagged values of M, as shown in Eqn. (4) below.

$$H_0: \quad S_t = \alpha + \beta_1 S_{t-1} + \beta_2 S_{t-2} + \ldots + \beta_i S_{t-i} + \varepsilon_t$$

$$H_1: \quad S_t = \alpha + \beta_1 S_{t-1} + \beta_2 S_{t-2} + \ldots + \beta_i S_{t-i} + \gamma_1 M_{t-1} + \gamma_2 M_{t-2} + \ldots + \gamma_j M_{t-j} + \varepsilon_t \qquad (4)$$

where index t refers to the time period (t = 1, …, T), i, j refer to the lags, and ε_t is supposed to be white-noise error. If the lagged values of M were individually significant according to their t-statistics, then we could reject the null hypothesis H_0 that there is no Granger-cause between M and S. In other words, the null hypothesis H_0 could not be rejected if there were no significant lagged values of M retained in the regression. We performed the GCA for the period between September 9, 2014 and March 24, 2015, with the exclusion of all non-trading days. All of the selected financial texts, with a total of 12 million words, were processed in our SAE using a dual Intel® Xeon CPU with 80GB RAM.

Endogenous variable	Exogenous variable	j = 0	j = 1	j = 2	j = 3	j = 4	j = 5
Daily Difference, S_t	M^+	0.2301	0.2634	0.8566	0.0397**	0.1945	0.4370
	M^-	0.0858*	0.5648	0.3297	0.0430**	0.7089	0.0307**
	M^{ND}	0.5114	0.9977	0.3723	0.4232	0.1822	0.0373**

Table 13: p-values of Granger causality correlation between different mood indices and the daily differences in the closing price of the stock market. * and ** indicate the significance levels at 10% and 5%, respectively. Note that this data set includes 866,628 sentences with 12 million words that were issued during the experiment period.

The exogenous variables M^+, M^- and M^{ND} stood for the positive, negative and net daily mood series. The coefficients and their statistical significance, or p-values, of the lagged values of the exogenous variables M^+, M^- and M^{ND} from j = 0 to 5 are shown in Table 13. The Granger causality test results are shown above. In general, according to the results of the tests (with 10% and 5% significance levels in the lagged exogenous variables) we could reject the null hypothesis that *the mood indices generated by the SAE do not Granger-cause any effect on stock index*. However, the effects of mood patterns on the stock market index varied considerably. The sentiments expressed seemed to take some time before gathering sufficient momentum to produce their influences on the market. The effects of the mood indices on the market at lagged values of j = 0 or 1 were not salient at the 5% significance level. However, the effects were significant at lagged values of j = 3 and 5. The sentiment latency effect was obvious, and the results showed that shifts in sentiment took at least three days to unfold and affect the market. In addition, it can be observed that the negative mood time series M^- had the highest Granger causality correlation with the daily difference in stock market closing price for j = 0 (p-value < 0.1), 3 and 5 (p-values < 0.05). A similar trend could also be found in the positive mood index M^+ and net daily mood index M^{ND}, even though the negative mood M^- showed a more longstanding effect on the market.

The last two evaluations on the financial text stream show that the mood indices generated from our SAE were not random series. They can predict the correct sentiment in financial texts and also hint on the changes in the stock market index, at least in the sense of Granger-cause. However, these results should be interpreted with at least two precautions. First, it was not the objective of this study to take advantage of investor sentiments for predicting the stock market index, nor to suggest any profitable trading rules. It is far too ambitious to claim that this method has any predictability. Even though the Granger concept of causality provides fundamental work for determining the relationships between time series variables, the Granger-cause based on temporal ordering is not necessarily true causality (Zellner, 1977). Market predictability relies on various factors, such as economic foundations, rates of interest, unemployment rates, market liquidity, spillover effects from other markets and many other factors. Further investigation is needed to reveal the importance of sentiments among other factors in predicting the market's directions. Second, the choice of the number of lags is a crucial step in the evaluation of texts, because the causality test results may depend critically on the number of lags chosen. In general, either too few or too many lags may be problematic. Important variables may be ignored in the test if the number of lags is insufficient. Such deficiency in the number of lags can introduce bias in the regression coefficients, as shown in (4), which can lead to incorrect conclusions. In contrast, more observations are consumed if a large lag length is chosen. With an overly large lag length, the sum of squared residuals is inflated, making the results inaccurate. Unfortunately, there is no silver bullet solution for determining the optimal number of lags. In the process of evaluation, the task of developing tags is conducted on a trial-and-error basis. We first start with a sufficiently larger number of lags and then fine-tune the number until the sum of squared residuals is significantly reduced. In summary, there is evidence that the mood indices are not a Brownian time series and that these indices provide hints concerning changes in the market index, at least in the sense of Granger-cause. These results provide grounds for belief that the sentiments generated through the financial text stream are helpful for analyzing the trends in financial markets.

VI. Conclusions

There are three main factors that distinguish our work from others. First, we have provided a novel approach to developing a language parser for sentiment analysis. This parser integrates various heterogeneous context features in refining the detection of chunking points and recognizing phrases by using an ensemble machine learning technique. A parse tree resolution algorithm is proposed to resolve the possible ambiguous or incomplete structures when multiple derivations are propagated from the terminal level. The sentiment parser is language-independent and can be easily adapted to different languages, such as in English and Chinese. Given the ever-evolving vocabularies in different languages, we suggest a way to do light parsing in a word-free context (rather than heavy parsing, which is highly dependent on word tokens) so that analysis can be done quickly without getting bogged down in various language genres. Second, building on the phrase structures rendered from the parse trees, we apply a sentiment assessment heuristic to assign the polarity of phrases. This heuristic also demonstrates how the polarity of a phrase can radiate up to its parents to derive the sentence-level polarity. A set of tactics for assessing the sentiments of texts at the sentence level is fully presented. As the results of testing indicate, the sentiment engine can process vast amounts of textual data at high speed without sacrificing meaning. This approach pushes beyond the keyword searches and string comparisons used in other forms of sentiment analysis. Third, to demonstrate that the engine is applicable to financial texts, we conduct several evaluations using movie review dataset and the data generated from financial text streams. Our statistical tests using twelve million words of financial text attest to the significance of our method. Our approach provides a means of identifying sentiments amid the complex and sometimes intimidating quantities of unstructured qualitative data derived from text media and other text-based repositories such as breaking news, news groups, blogs and tweets. Our proposed approach can help investors to deal with data in a way that eases demands on their cognitive limitations, but also reduces the negative effects of operating within bounded rationalities.

To consider all of the relevant factors for investment decision-making, it is important to formulate an impartial and well-justified method of assessing a vast flow of information. A diverse collection of independently deciding individuals is more likely to produce accurate predictions than any particular individual. The wisdom of the crowd concept, as suggested by Surowiecki (2005), involves winnowing the noise of individual judgments and aggregating them into a more comprehensive and accurate prediction. A process is necessary for taking the collective opinions of many individuals into account rather than relying on a single expert. Despite all of the infatuation with the wisdom of the crowd concept, the problem of how to devise a mechanism for turning individual judgments into comprehensive assessments still remains challenging. Our sentiment analysis provides a realization of this concept. Our approach offers snapshots of what the crowd (via millions of news announcements, blogs, tweets or other textual data) is thinking at a given instant. The results from applying this approach for evaluating financial texts and determining their overall statistical significance suggest that this method presents a feasible alternative for moving beyond the hype.

Acknowledgement

The authors are grateful to the editor and the anonymous referees for their valuable comments. Their constructive comments have already improved the paper from its previous version. The work described in this paper was partially supported by the grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Nos. CUHK440609 and CUHK440913).

References

Allen D.E., McAleer, M.J., Singh, A.K. (2015). Machine news and volatility: The Dow Jones Industrial Average and the TRNA real-time high-frequency sentiment series. In G.N. Gregoriou (Ed.), *The Handbook of High Frequency Trading*, 327-344. Academic Press.

Antweiler, W., & Frank, M. Z. (2004). Is all that talk just noise? The information content of internet stock message boards. *Journal of Finance*, 59, 1259-1294.

Baccianella, S., Esuli, A., & Sebastiani, F. (2010). SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In N. Calzolari, and K. Choukri, B. (Eds.), *LREC*: European Language Resources Association.

Baker, M., & Wurgler, J. (2006). Investor sentiment and the cross-section of stock returns. *Journal of Finance*, 61, 1645-1680.

Bikel, D. M. (2004). *On the Parameter Space of Generative Lexicalized Statistical Parsing Models*. PhD thesis, University of Pennsylvania.

Billot, S., & Lang, B. (1989). The structure of shared forests in ambiguous parsing. *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, 143-151.

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood as a stock market predictor. *Computers and Operations Research*, 44, 91-94.

Cecchini, M., Aytug, H., Koehler, G. J., & Pathak, P. (2010). Making words work: Using financial text as a predictor of financial events. *Decision Support Systems*, 50, 164-175.

Chan, S.W.K., & Chong, M.W.C. (2013). Recursive part-of-speech tagging using word structures. *Lecture Notes in Artificial Intelligence*, v.8082, 419-425, Springer-Verlag.

Chan, S.W.K., Chong, M. W. C., & Cheung, L. Y. L. (2011). An analysis of tree topological features in classifier-based unlexicalized parsing. *Lecture Notes in Computer Science*, v. 6608, 155-170, Springer-Verlag.

Chan, W.S. (2003). Stock price reaction to news and no-news: Drift and reversal after headlines. *Journal of Financial Economics*, 70, 223-260.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No.3, Article 27.

Chen, H., Chiang, R. H. L., & Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. *MIS Quarterly*, 36, 4, 1165-1188.

Chen, K.-T., Lu, H.-M., Chen, T.-J., Li, S.-H., Lian, J.-S., & Chen, H. (2011). Giving context to accounting numbers: The role of news coverage. *Decision Support Systems*, 50, 673-679.

Collins, M. (1999). *Head-driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia.

Couillard, M., & Davison, M. (2005). A comment on measuring the Hurst exponent of financial time series. *Physica A*, 348, 404-418.

Ederington, L.H., & Lee, J.H. (1993). How markets process information: News releases and volatility. *Journal of Finance*, 48, 1161-1191.

Eickhoff, M., & Muntermann, J. (2016). Stock analysts vs. the crowd: Mutual prediction and the drivers of crowd wisdom. *Information and Management*.

Engle, R.F., & Ng, V.K. (1993). Measuring and testing the impact of news on volatility. *Journal of Finance*, 48, 1749-1778.

Fersini, E., Messina, E., & Pozzi, F.A. (2014). Sentiment analysis: Bayesian ensemble learning. *Decision Support Systems*, 68, 26-38.

Garcia, D. (2013). Sentiment during recessions. *Journal of Finance*, 68, 3, 1267-1300.

Granger, C.W.J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37, 3, 424-438.

Groth, S.S., & Muntermann, J. (2011). An intraday market risk management approach based on textual analysis. *Decision Support Systems*, 50, 680-691.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. *Proceedings of the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Seattle.

Huang, C. -L., Chung, C. K., Hui, N., Lin, Y. -C., Seih, Y. -T., Chen, W. -C., Lam, B., Bond. M., & Pennebaker, J. W. (2012). The development of the Chinese Linguistic Inquiry and Word Count Dictionary. *Chinese Journal of Psychology*, 54, 2, 185-201 (in Chinese).

Hurst, H.E. (1951). Long-term storage of reservoirs: An experimental study. *Transactions of the American Society of Civil Engineers*, 116, 770-799.

Joachims, T. (2002). *Learning to Classify Text Using Support Vector Machines*. Kluwer.

Klein, F., & Prestbo, J. A. (1974). *News and the Markets*. Henry Regnery, Chicago.

Kothari, S., Li, X., & Short, J. (2009). The effect of disclosures by management, analysts, and business press on cost of capital, return volatility, and analyst forecasts: a study using content analysis. *Accounting Review*, 84, 1639–1670.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool.

Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *Journal of Finance*, 66, 35-65.

Loughran, T., & McDonald, B. (2013). IPO first-day returns, offer price revisions, volatility, and form S-1 language. *Journal of Financial Economics*, 109, 307-326.

Lu, Y., Zhai, C., & Sundaresan, N. (2009). Rated aspect summarization of short comments. *Proceedings of 18th International World Wide Web Conference (WWW'09)*, Madrid, Spain.

Mandelbrot, B. B. (2004). *The (Mis)Behavior of Markets: A Fractal View of Risk, Ruin and Reward*. Basic Books.

Marcus, M., Santorini, M., & Marcinkiewicz, M. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19, 2, 313-330.

Melvin, M., & Yin, X. (2000). Public information arrival, exchange rate volatility and quote frequency. *Economic Journal*, 110, 644-661.

Mitra, G., & Mitra, L. (2011). *The Handbook of News Analytics in Finance*. West Sussex: John Wiley.

Palmer, M., Chiou, F.-D., Xue, N., & Lee, T.-K. (2005). *Chinese Treebank 5.0* LDC2005T01. Web Download. Linguistic Data Consortium: Philadelphia.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*

Pierce, D. A. (1977). Relationships and lack thereof between economic time series, with special reference to money and interest rates. *Journal of the American Statistical Association*, 72 (March), Applications Sections, 11-22.

Poon, S.-H., & Granger, C. W. J., (2005). Practical issues in forecasting volatility. *Financial Analysts Journal*, 61, 1, 45-56.

Quinlan, R. (1994). *C4.5: Programs for Machine Learning.* Morgan Kaufmann.

Rosenbach, A. (2005). Animacy versus weight as determinants of grammatical variation in English. *Language*, 81, 3, 613-644.

Schumaker, R. P., & Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news: The AZFin text system, *ACM Transactions on Information Systems*, 27, 2, Article 12.

Schumaker, R.P., Zhang, Y., Huang, C. & Chen, H. (2012). Evaluating sentiment in financial news articles. *Decision Support Systems*, 53, 3, 458-464

Shaikh, M. A. M., Prendinger, H., & Mitsurs, I. (2007). Assessing sentiment of text by semantic dependency and contextual valence analysis. *Proceedings of the International Conference on Affective Computing and Intelligent Interaction (ACII-07)*, 191-202.

Shiller, R. J. (2000). *Irrational Exuberance*. Princeton University Press.

Sun, H., & Jurafsky, D. (2004). Shallow semantic parsing of Chinese. *Proceedings of NAACL-HLT*.

Surowiecki, J. (2005). *The Wisdom of Crowds*. Anchor.

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *Journal of Finance*, 62, 1139-1168.

Todorovski, L., & Džeroski, S. (2003). Combining classifiers with meta decision tress. *Machine Learning Journal*, 50, 3, 223-249.

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

Wang, G., Sun, J., Ma, J., Xu, K., & Gu, J. (2014). Sentiment classification: The contribution of ensemble learning. *Decision Support Systems*, 57, 77-93.

Wasow, T. (1997). Remarks on grammatical weight. *Language Variation and Change*, 9, 81-105.

Wei, W., & Gulla, J. A. (2010). Sentiment learning on product reviews via sentiment ontology tree. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 404-413.

Wiebe, J., & Mihalcea, R. (2006). Word sense and subjectivity. *Proceedings of ACL-06*, 1065-1072.

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. *Proceedings of HLT/EMNLP*, 347-354.

Wuthrich, B., Permunetilleke, D., Leung, S., Lam, W., Cho, V., & Zhang J. (1998). Daily predication of major stock indices from textual WWW data. *HKIE Transactions*, 5, 3, 151-156.

Zellner, A. (1977). Comments on time series and causal concepts in business cycle research. *New Methods in Business Cycle Research*, ed. C.A. Sims. Federal Reserve Bank of Minneapolis.

Ženko, B., Todorovski, L., & Džeroski, S. (2001). A comparison of stacking with meta decision trees to other combining methods. *Proceedings of the Fourth International Multi-Conference Information Society*, vol. A., 144-147. Jozef Stefan Institute, Ljubljana.

Samuel Chan received the M.Sc. degree in 1986 from the University of Manchester, U.K., and M.Phil. degree in 1991 from the Chinese University of Hong Kong, and the Ph.D. degree in 1998 from the University of New South Wales, Australia, all in Computer Science. Before joining the Chinese University of Hong Kong as an associate professor, he has been working in computational intelligence since 1989. His current research interests are in applying machine learning techniques in text, text mining, information retrieval and natural language processing with emphasis on text-based decision making. He had won the best paper award among 480 entries from 37 countries at the *7th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2006)* as well as the first place in one of the tasks in *Fourth International Chinese Language Processing Bakeoff* (2008). He has published articles in *Decision Support Systems*, *IEEE Transactions on Neural Networks, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Systems, Man, and Cybernetics, Journal of Information Science, Machine Translation, Journal of Chinese Linguistics, Artificial Intelligence in Medicine, Expert Systems with Applications, Applied Artificial Intelligence, International Journal of Computer Processing of Oriental Languages* and others.

Mickey Chong received the M.Sc. degree in Computer Science from the Chinese University of Hong Kong. He is a research associate at the Department of Decision Sciences in the University, with research interest in text processing and computational linguistics. He is one of the chief architects designing various internet-based applications in text analysis.

Highlights of the Paper

- To explain a classifier-based sentiment parser for financial texts
- To demonstrate how to assign the polarity of phrases using an assessment heuristic
- To provide statistical tests using twelve million words to attest its significance