# The Impact of Word Sense Disambiguation on Stock Price Prediction

Alexander Hogenboom, Alex Brojba-Micu, Flavius Frasincar*

*Erasmus University Rotterdam, P.O. Box 1738, NL-3000 DR Rotterdam, the Netherlands*

**Abstract**

State-of-the-art decision support systems for stock price prediction incorporate pattern-based event detection in text into their predictions. These systems typically fail to account for word meaning, even though word sense disambiguation is crucial for text understanding. Therefore, we propose an advanced natural language processing pipeline for event-based stock price prediction, that allows for word sense disambiguation to be incorporated in the event detection process. We identify events in natural language news messages and subsequently weight these events for their historical impact on stock prices. We assess the merit of word sense disambiguation in event-based stock price prediction in two evaluation scenarios for NASDAQ-100 companies, based on historical stock prices and news articles retrieved from Dow Jones Newswires over a 2-year period. We evaluate the precision of generated buy and sell signals based on our predicted stock price movements, as well as the excess returns generated by a trading strategy that acts upon these signals. Event-based stock price predictions seem most reliable about 2 days into the future. The number of detected events tends to reduce with over 30% when graph-based word sense disambiguation using a degree centrality measure is applied in the event detection process, thus reducing the noise introduced into the stock price movement predictions by high-impact ambiguous events. As a result, modest improvements in the precision of buy and sell signals generated based on these predictions tend to lead to vast improvements of on average about 70% in the associated excess returns.

*Keywords:* Stock price prediction, Event detection, Word sense disambiguation, Natural language processing

*Corresponding author; tel: +31 (0)10 408 1340; fax: +31 (0)10 408 9162

*Email addresses:* hogenboom@ese.eur.nl (Alexander Hogenboom), 289585ab@student.eur.nl (Alex Brojba-Micu), frasincar@ese.eur.nl (Flavius Frasincar)

## 1. Introduction

Electronic word-of-mouth phenomena and speculative bubbles have repeatedly demonstrated the extent to which today's markets are affected by people's moods and opinions. For instance, exchange rates of cryptocurrencies like Bitcoin have been shown to be extremely sensitive to the craze of the day (Kristoufek, 2013). Furthermore, reputation (Jansen et al., 2009), sales (Yu et al., 2012; Rui et al., 2013), and stock ratings (Yu et al., 2013; Ho et al., 2017) tend to be significantly influenced by subjective tweets, reviews, and other social media content. In stock markets, not only subjective user-generated content (Yu et al., 2013; Ho et al., 2017) and rumors (van Bommel, 2003), but also arguably more objective news messages (Chan, 2003; Zhang, 2006) reporting on relevant events have been shown to affect both trading volume and stock prices.

Stocks have been shown to exhibit abnormal returns once news messages are made publicly available (Chan, 2003). The extent and nature of the impact of such news depends on the degree of uncertainty around an asset (Zhang, 2006), as well as on the nature of the news messages and the events they report on. For one, trading activity tends to increase with the number of news messages (Mitchell & Mulherin, 1994). Additionally, merger announcements for companies with successful past mergers tend to positively influence the stock price momentum (Rosen, 2006). Observations like these warrant a need to identify relevant events, and to account for their effects when trading on stock markets.

The abundance of data in this era of Big Data can be used to monitor the wants, the needs, and the opinions of many stakeholders with respect to a topic of interest (Hogenboom et al., 2015a). Targeted data analysis enables decision makers to identify issues and patterns that matter, and to track and predict emerging events (Hogenboom et al., 2014). These types of analysis are no longer limited to structured data, but tend to include more and more unstructured data like natural language text as well (Montoyo et al., 2012; Hogenboom et al., 2015a,b). The typical focus here is on retrieving explicit pieces of information from text, on varying levels of granularity (Chang et al., 2006). This enables the identification of entities like companies, products, or brands in text, and the subsequent extraction of more complex concepts, such as events in which these entities play distinct roles (Hogenboom et al., 2013, 2016).

Interestingly, even though state-of-the-art decision support systems already allow us to identify events in natural language text, such events are yet to be exploited to their full extent in stock price prediction. State-of-the-art stock price prediction methods are primarily statistics-based approaches (Roh, 2007; Lu, 2010; Kara et al., 2011; Ariyo et al., 2014; Patel et al., 2015), with some approaches incorporating some basic text analysis (Fawcett & Provost, 1999; Lavrenko et al., 2000; Peramunetilleke & Wong, 2002; Hagenau et al., 2013; de Fortuny et al., 2014), sentiment analysis (Ahmad et al., 2002; Bollen et al., 2011; Schumaker et al., 2012; Feuerriegel & Prendinger, 2016), or pattern-based event detection (Seo et al., 2002; Nuij et al., 2014) into the process. Word Sense Disambiguation (WSD) is arguably one of the most crucial steps towards text understanding (Navigli & Velardi, 2005). Nevertheless, accounting for word meaning in stock price prediction is not well-researched, with a notable exception being modeling local word context by using two-word features (Hagenau et al., 2013).

In this light, we propose to assess the impact of WSD on stock price prediction. The main contribution of our work lies in an advanced natural language processing pipeline for event-based stock price prediction, that allows for a crucial WSD step to be incorporated in the process of identifying events that can affect stock prices. Our stock price prediction method identifies events in natural language news messages and subsequently weights these events for their associated impact on stock prices. This impact stems from a statistical analysis that captures the contribution of individual events to changes in historical stock prices. We show that both stock price prediction precision and excess returns on the stock market are positively affected when enriching the event extraction process with WSD.

The remainder of this paper is structured as follows. First, we discuss the state-of-the-art in stock price prediction and WSD in Section 2. Then, we propose and evaluate our pipeline for event-based WSD-enabled stock price prediction in Sections 3 and 4, respectively. Last, we present our conclusions and directions for future work in Section 5.

## 2. Related Work

The state-of-the-art in stock price prediction encompasses a large variety of approaches. Many of these approaches, as discussed in Section 2.1, rely on textual data. A crucial yet typically overlooked step in text-based stock price prediction is to identify the intended senses of ambiguous words. Methods for doing so are discussed in Section 2.2.

### *2.1. Stock Price Prediction*

State-of-the-art stock price prediction methods are primarily statistics-based (Roh, 2007; Lu, 2010; Kara et al., 2011; Ariyo et al., 2014; Patel et al., 2015). Some methods use basic text analysis (Fawcett & Provost, 1999; Lavrenko et al., 2000; Peramunetilleke & Wong, 2002; Hagenau et al., 2013; de Fortuny et al., 2014) or sentiment analysis (Ahmad et al., 2002; Bollen et al., 2011; Schumaker et al., 2012; Feuerriegel & Prendinger, 2016) into the process. The most advanced approaches apply a form of pattern-based event detection (Seo et al., 2002; Nuij et al., 2014).

#### *2.1.1. Statistical Analyses*

As stock price movements over time are essentially time series, some existing work approaches stock price prediction, especially in short-term scenarios, as a time series prediction problem. For example, Ariyo et al. (2014) successfully apply autoregressive integrated moving average models in order to predict short-term stock price movements on the New York Stock Exchange (NYSE) and Nigeria Stock Exchange (NSE). Other work forecasts stock price index movements by means of hybrid models that combine time series analyses with, e.g., artificial neural networks (Roh, 2007).

In recent years, machine-learning models like artificial neural networks have been shown to be very effective in stock price prediction scenarios. For example, Kara et al. (2011) predict the direction of movement in the daily Istanbul Stock Exchange (ISE) National 100 Index based on ten technical indicators by means of artificial neural networks and support vector machines, with the artificial neural networks yielding significantly better results than the support vector machines. The performance of machine-learning models can be further improved by preprocessing their inputs (Lu, 2010; Patel et al., 2015). For instance, Lu (2010) reconstructs forecasting variables by means of an integrated independent component analysis in order for them to contain less noise, and subsequently feeds these variables into an artificial neural network in order to forecast stock prices of the TAIEX and Nikkei 225 index. Patel et al. (2015) predict movements of stock and stock price index for two Indian stocks and two Indian stock price indices by means of artificial neural networks, support vector machines, random forests, and naïve Bayes classifiers. Patel et al. (2015) show that the performance of these models can be improved by preprocessing the technical indicators that serve as input for these models by representing them as trend deterministic data.

#### *2.1.2. Text Analytics*

In recent work, de Fortuny et al. (2014) provide an extensive overview of how text analytics are exploited in state-of-the-art stock price prediction models, that typically focus on predicting the direction of stock price movements. In light of this meta-analysis, de Fortuny et al. (2014) propose a hybrid stock price prediction model. This model is a support vector machine that combines technical indicators with simple bag-of-word representations of the title, the most relevant paragraph, and the full text of news messages, as well as with the sentiment associated with these messages. The model is evaluated on various performance metrics, and de Fortuny et al. (2014) note that the observed performance improvements over less intricate stock price prediction approaches vary across those metrics. This implies that stock price prediction models that include text analytics are more suitable for decision support tools than they are for decision making tools – human judgment remains crucial when analyzed text is ambiguous, especially when it is not or poorly disambiguated.

Nevertheless, many existing approaches steer clear from a hybrid approach, and attempt to maximize the benefits of text analytics in stock price prediction. For example, Fawcett & Provost (1999) present a framework that monitors news messages for keywords signalling a positive trend, and as such predicts positive spikes in stock prices. Similarly, Ahmad et al. (2002) attempt to predict stock price trends based on specific keywords in news messages. Ahmad et al. (2002) show that changes in the frequency of positive words in news messages correlate well with changes in the Financial Times Stock Exchange (FTSE) 100 index. Similarly, Peramunetilleke & Wong (2002) show that currency exchange rate movements can be predicted by using probabilistic rules that account for keywords in news headlines.

More advanced approaches do not focus on specific keywords, but rather on the text as a whole when predicting trends in financial markets. For example, Lavrenko et al. (2000) correlate the content of news stories with trends in financial time series, and then identify relevant news messages that are highly indicative of future trends. Lavrenko et al. (2000) demonstrate that a Bayesian model can profitably predict stock prices based on the recommended news messages. In other work, Hagenau et al. (2013) make a first attempt at accounting for word meaning when predicting stock prices based on the text of relevant news messages. Their bag-of-words representation of news messages includes two-word features that aim to capture the local context of words. Bollen et al. (2011) demonstrate that the movements of the Dow Jones Industrial Average (DJIA) index can be predicted based on public moods, mined from a Twitter feed. Similarly, Schumaker et al. (2012) and Feuerriegel & Prendinger (2016) incorporate the sentiment associated with relevant news messages into a machine-learning model that predicts price movements for the associated stocks. They demonstrate how accounting for the sentiment conveyed by relevant text can significantly improve the effectiveness of stock trading strategies in terms of profits, but at the cost of risk.

#### *2.1.3. Event Detection*

Rather than using individual words, their context, and their conveyed sentiment as direct proxies for stock price movements, some approaches use such textual cues in order to detect relevant events that are described in text. These detected events are subsequently used in order to predict stock price movements or to make buy or sell decisions on stock markets. An extensive survey of event detection in general was done by Hogenboom et al. (2016).

An early example of an application of event detection in financial markets is Warren, i.e., an ensemble of intelligent agents, designed to assist humans in financial markets (Seo et al., 2002). One of the tasks performed by Warren is to classify news messages about companies of interest as either positive facts, positive forecasts, neutral, negative forecasts, or negative facts. Warren classifies news messages by using a naïve Bayes classifier that focuses on frequently collocated phrases in sentences that contain the name of a company of interest.

In later work, StockWatcher (Micu et al., 2008) was introduced. This application aims to aggregate news messages about companies, their competitors, their most important employees, and their industry. Additionally, StockWatcher classifies the effects of the events described by these news messages on stock prices as either positive, neutral, or negative. The system relies on an ontology with information about industries, companies, people, and economic events. Furthermore, StockWatcher uses pre-defined heuristics in order to quantify the impact of the economic events in the news messages on stock prices.

In more recent work, Nuij et al. (2014) detect financial events in large amounts of news messages and subsequently use these events in order to determine whether to buy or sell stocks for companies of interest. Their system relies on a domain-specific ontology that contains properties and lexical representations of companies. This ontology is used in order to detect relevant concepts in news messages. These concepts are subsequently mapped onto relevant events by applying semantic, morphological, syntactical, and typographical patterns. By means of a genetic programming approach, Nuij et al. (2014) combine the events thus detected with technical indicators into trading rules that determine whether to buy or sell stocks.

### *2.2. Word Sense Disambiguation*

With many of the state-of-the-art stock price prediction models discussed in Section 2.1 (partly) relying on textual data, dealing with ambiguity in natural language is an essential, but to the best of our knowledge typically overlooked step in the analysis. In order to be able to reason with the information in text, one must accurately understand this information. Identifying the correct senses of the words in the context in which they are used, i.e., Word Sense Disambiguation (WSD) is one of the most crucial steps towards text understanding (Navigli & Velardi, 2005).

Automated WSD has applications in many, disparate natural language processing tasks, where it typically serves as an intermediary step towards a specific goal. Notable applications can be found in machine translation (Brown et al., 1991; Hutchins & Somers, 1992), query-driven information retrieval (Krovetz & Croft, 1992; Sanderson, 1994), and numerous information extraction and knowledge acquisition applications like sentiment analysis (Heerschop et al., 2011a,b; Hogenboom et al., 2014, 2015b), the uncovering of a social network of people in biographical texts (van de Camp & van den Bosch, 2012), the extraction of a domain taxonomy from a collection of texts (Meijer et al., 2014), and the extraction of tips for product improvements from reviews (Zhu et al., 2018). Nowadays, WSD is typically considered to be a largely solved problem that can be tackled in various ways, with most approaches being either corpus-based (Section 2.2.1) or knowledge-based (Section 2.2.2).

### *2.2.1. Corpus-Based Approaches*

The premise of corpus-based approaches to WSD is that large collections of text provide sufficient examples of words in various contexts, and that machine-learning classifiers can use these examples in order to identify the senses of words in similar pieces of text based on the context of these words. Typical training methods for such classifiers are supervised, semi-supervised, or unsupervised.

The training and evaluation of corpus-based WSD approaches requires large collections of pieces of text, where ambiguous words are annotated with their correct senses. Two of such widely used corpora are SemCor (Miller et al., 1993) and Senseval (ACL-SIGLEX, 1998), both of which encode word senses as distinct concepts in WordNet (Fellbaum, 1998). WordNet is a vast lexical resource that is organized into sets of cognitive synonyms, i.e., synsets that can be differentiated based on their part-of-speech (POS) type. Each out of the 117,659 synsets in WordNet expresses a distinct concept and is linked to other synsets through various types of relations like synonymy, antonymy, hyponymy, or meronymy. SemCor is the largest publicly available sense-tagged corpus. It is composed of 352 documents extracted from the Brown Corpus (Francis & Kucera, 1964), which is a million-word balanced collection of English texts that were published in the United States in 1961. In SemCor, for 186 of these documents, all of the 192,639 nouns, verbs, adjectives, and adverbs are annotated with their part-of-speech (POS), lemma, and WordNet synset, whereas for 166 documents only 41,497 verbs are annotated with their POS, lemma, and WordNet synset. Senseval was a series of workshops (Kilgarriff, 1998; Edmonds & Cotton, 2001; Mihalcea & Edmonds, 2004) where WSD systems were evaluated on specific WSD tasks, and the data for these tasks is still available.

Supervised machine-learning approaches to WSD require large amounts of annotated data from corpora like SemCor and Senseval in order to train a classifier. Some of these approaches represent all annotated training instances in memory and then compare each new and unclassified instance to the ones in memory in order to determine its most likely word sense (Ng, 1997a; Fuji et al., 1998; Hoste et al., 2002; Decadt et al., 2004). Other methods aim to distill a set of (hierarchical) rules from the training instances and then use these rules in order to determine the word senses of unseen instances (Kelly & Stone, 1975; Black, 1988; Mooney, 1996; Yarowsky, 1994, 2000; Agirre & Martinez, 2000; Pedersen, 2001). Other approaches involve statistical models that maximize the conditional probabilities of the senses of words given their context using, e.g., naïve Bayes classifiers (Mooney, 1996; Ng, 1997b; Leacock et al., 1998; Bruce & Wiebe, 1998; Le & Shimazu, 2004). Support vector machines (Lee & Ng, 2002; Lee et al., 2004; Joshi et al., 2005; Buscaldi et al., 2006) and neural networks (Cottrell, 1989; Veronis & Ide, 1990; Chung et al., 2002; Lu et al., 2004; Tsatsaronis et al., 2007; Iacobacci et al., 2016) have been shown to be well-performing supervised machine-learning approaches to WSD as well.

The downside of supervised machine-learning methods is that they require large amounts of annotated data. Semi-supervised methods like bootstrapping (Hearst, 1991; Yarowsky, 1995; Mihalcea & Moldovan, 2001) mitigate this concern by using only a small seed set of annotated data in order to train an initial WSD classifier. This initial classifier is used to automatically annotate a large amount of data. The samples that are annotated with the highest confidence are then used as additional data for (iteratively) retraining the WSD classifier. Unsupervised clustering-based approaches (Pedersen & Bruce, 1997; Niu et al., 2004; Ji, 2010) have been proven to be effective methods too.

*2.2.2. Knowledge-Based Approaches*

Alternative WSD approaches exploit representations of real-world knowledge with respect to the senses of words in a specific context, rather than obtaining these insights from sense-tagged corpora. Some of these approaches attempt to match the context of ambiguous words with descriptions of senses in dictionaries (Lesk, 1986; Cowie et al., 1992). Other approaches exploit thesauri, capturing synonyms and antonyms in semantic categories that serve as a proxy for word senses (Walker, 1987; Yarowsky, 1992). The application of dictionaries and thesauri has had limited success, since their lack of pragmatic information renders dictionaries and thesauri primarily useful for use by humans rather than by machines (Ide & Veronis, 1998). Conversely, lexical repositories like WordNet

(Fellbaum, 1998) capture lexical representations, descriptions, and relations of words in a way that is optimized for machine-interpretability of semantic categories, and have as such been shown to be very useful in WSD.

One intuitive way of disambiguating the sense of a word using a lexical repository like WordNet is to pick the semantic category of the sense that is closest to the semantic categories that represent the local context of the word. This distance between two semantic categories can be measured in terms of the length of the shortest path of hierarchical is-a and part-of relations between these categories (Rada et al., 1989; Patwardhan et al., 2003). In order to account for the specificity of deeper semantic categories, the path-based distance can be decreased proportionally to the depth of the semantic categories in the hierarchy (Leacock & Chodorow, 1998). Moreover, in addition to the vertical is-a and part-of relations, horizontal relations like synonym and antonym relations can be accounted for as well, by favoring shorter paths with few direction changes over longer paths with many direction changes (Hirst & St-Onge, 1998). An alternative way of measuring the distance between two semantic categories is to measure the length of the path to their nearest, most specific shared ancestor, i.e., the least common subsumer (Wu & Palmer, 1994).

Another way of disambiguating word senses using a lexical repository is to account for the specificity of semantic categories by quantifying their information content. This information content is typically inversely proportional to the likelihood of a semantic category to occur in a text. Resnik (1995) proposes to use the information content of the least common subsumer of two semantic categories as a proxy for their similarity – the more specific the least common subsumer, the more similar the word senses are. Thus, the information content of the least common subsumer can be used to identify the semantic category of the word sense that is most similar to the semantic categories of the local context of a word. In this process, one can additionally account for the discrepancy between the information content of the actual semantic categories of the local context and of those semantic categories representing the potential word senses (Jiang & Conrath, 1997; Lin, 1997). The algorithm of Jiang & Conrath (1997) has been shown to be one of the most effective WSD algorithms based on path length and/or information content (Patwardhan et al., 2003).

The aforementioned WSD methods guided by path-based semantic similarity or by information content focus on disambiguating the senses of individual words one by one. Conversely, another class of WSD algorithms disambiguates all senses in a piece of text collectively – the more computationally intensive WSD algorithms that rely on graph-based similarities between word senses (Mihalcea, 2005; Navigli & Lapata, 2007, 2010; Sinha & Mihalcea, 2007; Amancio et al., 2012; Koppula et al., 2017; Correa Jr. et al., 2018; Correa Jr. & Amancio, 2019). Such graph-based methods typically consist of three steps (Sinha & Mihalcea, 2007). First, all possible word senses are represented as nodes in a graph, with edges between them signalling (the extent of) interdependencies that are derived from annotated data or lexical repositories. Second, all of these senses are assigned a score, based on their centrality in the graph. Last, for each ambiguous word, the sense with the highest graph centrality score is selected.

The interdependencies between word senses can be derived from their similarities, as quantified by the length of the shortest path between them (Hirst & St-Onge, 1998). This can be combined with the information content of these senses as well – for example, Sinha & Mihalcea (2007) propose an algorithm that uses the Jiang & Conrath (1997) algorithm for nouns, the Leacock & Chodorow (1998) algorithm for verbs, and the Lesk (1986) algorithm for adjectives and adverbs. Alternatively, the overlap in terms of the number of tokens that word senses have in common can serve as a proxy for their interdependencies (Mihalcea, 2005). Another method is to simply use a lexical repository like WordNet to identify how all senses of the words in a piece of text are directly or indirectly connected (Navigli & Lapata, 2010).

Graph centrality scoring algorithms come in a variety of shapes and forms. These algorithms are typically either global or local algorithms. Global graph centrality scoring algorithms evaluate and attempt to maximize the overall connectivity of the graph of a text's selected word senses. Conversely, local graph centrality scoring algorithms evaluate the connectivity of individual word senses in a graph, and select those word senses that have the highest connectivity. Such local algorithms tend to outperform global ones (Navigli & Lapata, 2007, 2010). A popular local graph connectivity measure is the degree centrality (Freeman, 1979), which in its simplest form captures the number of connections a word sense has with other word senses in the graph. PageRank

(Brin & Page, 1998) and eigenvector centrality measures (Bonacich, 1972) are more sophisticated variants of this, that assign more weight to connections to highly connected word senses. Degree centrality measures have been shown to be the best performing measures for WSD purposes (Sinha & Mihalcea, 2007; Navigli & Lapata, 2010). Other centrality measures account for the distance between word senses in the graph. For instance, the closeness centrality (Sabidussi, 1966) of a word sense is inversely proportional to the total shortest distance from a word sense to all other word senses in the graph, thus favoring word senses that are closest to all other word senses. Another distance-based centrality measure for a word sense is the betweenness centrality (Freeman, 1979), which captures the fraction of shortest paths between word senses in the graph that pass through the former word sense.

## 3. Word Sense Disambiguation in Event-Based Stock Price Prediction

With the most advanced stock price prediction approaches relying on event detection, yet failing to account for the intended senses of ambiguous words, we propose StockWatcher 2.0 – an advanced natural language processing pipeline for event-based stock price prediction, that allows for a crucial WSD step to be incorporated in the process of identifying events that can affect stock prices. Our approach identifies events in natural language news messages and subsequently weights these events for their associated impact on stock prices when forecasting stock price movements.

StockWatcher 2.0 builds upon our previous work, i.e., StockWatcher (Micu et al., 2008). The initial version of StockWatcher identifies economic events described in news messages about companies, their competitors, their most important employees, and their industry. StockWatcher then uses pre-defined heuristics in order to quantify the impact of these events on stock prices. In StockWatcher 2.0, we replace these heuristics with a statistical analysis that captures the contribution of individual economic events to changes in historical stock prices. Moreover, StockWatcher 2.0 extends StockWatcher by incorporating WSD in the event detection process. Figure 1 visualizes our event-based stock price prediction process with support for WSD.

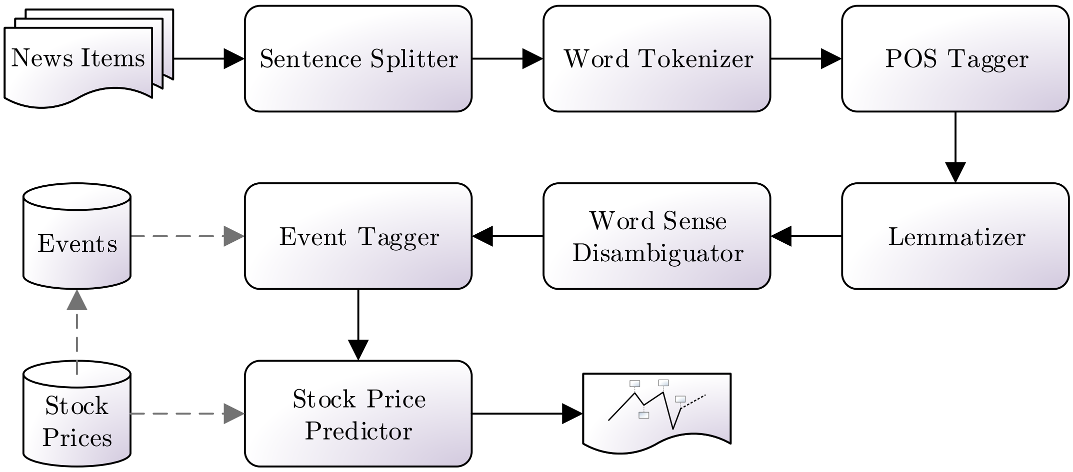


Figure 1: Overview of our event-based stock price prediction framework. Solid arrows signal the information flow, whereas dashed arrows indicate a used-by relationship.

### *3.1. Inputs*

StockWatcher 2.0 has three distinct input sources. The first source is a collection of historical daily stock opening and closing prices, as well as daily index opening and closing prices. In this work, we focus this collection on NASDAQ-100 listed companies and hence the associated NASDAQ index. The second source consists of news items that are relevant to the companies of interest. StockWatcher 2.0 ingests these new messages via RSS feeds and categorizes them by company by means of named entity recognition based on known lexical representations of the considered companies. The third input source is a knowledge base with representations of economic events. Each event is represented by a set of lemmas (i.e., the canonical forms) of the nouns, verbs, adjectives, and adverbs that typically describe it. For events that require disambiguation, we store the POS and WordNet senses of the lemmas as well. Additionally, we store the impact over time – i.e., 0, 2, and 4 days – of each event on each of the considered companies.

We compute the impact of events on stock prices using a formula that is inspired by the field of sentiment analysis, where Cesarano et al. (2006) determine the contribution of individual words to an opinion by means of a pseudo-expected value computation. Similarly, we compute the contribution of individual events to changes in stock prices. We model the impact $w_{e_c t}$ of event

$e_c$ for company $c$ on its associated stock price, $t$ days after day $d$ of the event's occurrence, as the weighted average of the daily relative importance (i.e., occurrence rate or prevalence) of this event, with the daily weights being the associated stock price changes $t$ days into the future, i.e.,

$$w_{e_c t} = \frac{\sum_{d \in D} \left( \frac{|e_{cd}|}{\sum_{e'_{cd} \in E_{cd}} |e'_{cd}|} \cdot \frac{p'_{c(d+t)} - p_{cd}}{p_{cd}} \right)}{\sum_{d \in D} \left( \frac{p'_{c(d+t)} - p_{cd}}{p_{cd}} \right)}, \tag{1}$$

with day $d$ in all considered days $D$ in the historical observations, $|e_{cd}|$ the total frequency of event $e$ for company $c$ on day $d$, $E_{cd}$ all events for company $c$ on day $d$, $p_{cd}$ the opening stock price of company $c$ on day $d$, and $p'_{c(d+t)}$ the closing stock price of company $c$ on day $d + t$.

### *3.2. News Message Processing*

In order to detect economic events in news messages, StockWatcher 2.0 subjects these news messages to a series of text processing steps. First, the text is split into sentences and words by means of regular expressions. Then, StockWatcher 2.0 determines the POS and lemma of each resulting word. The word senses of ambiguous lemmas are subsequently identified by means of a graph-based WSD component that implements the approach of Sinha & Mihalcea (2007). We have selected this algorithm over the others discussed in Section 2.2 as it is an unsupervised algorithm that has been shown to have superior performance to other unsupervised WSD algorithms in the past (Sinha & Mihalcea, 2007) and that does not require as much training data as alternative approaches like those based on word embeddings do (Iacobacci et al., 2016). As such, for each ambiguous lemma, a graph with all of the possible senses of the other lemmas with the same POS in the sentence is created, and their interdependencies are scored with the Jiang & Conrath (1997) algorithm for nouns, the Leacock & Chodorow (1998) algorithm for verbs, and the Lesk (1986) algorithm for adjectives and adverbs (all words with the same POS are disambiguated together). This in turn allows for the sense with the highest degree centrality score (Freeman, 1979) to be selected. The resulting lemmas and, where applicable, disambiguated word senses are then matched against the lemmas and their associated word senses in the knowledge base with economic events in order to identify events. The latter step is only performed for sentences that mention the company of interest, as well as the sentences that precede or succeed these sentences, as the events in these sentences are most likely relevant for the company of interest (Micu et al., 2008).

### *3.3. Outputs*

StockWatcher 2.0 produces two different types of output. First, it produces a chronological overview of news items for a company of interest, along with a chart that displays the observed historical stock prices for this company. Second, StockWatcher 2.0 predicts the relative change $\tilde{p}_{cdt}$ in the stock price of a company $c$, in a time window of $t$ days between day $d$ and $d + t$, based on the sum of each detected company-specific event weight $w_{e_c t}$ for event $e_c$ out of all detected events $E_{cd}$ for company $c$ on day $d$ on the associated stock price, i.e.,

$$\tilde{p}_{cdt} = \sum_{e_c \in E_{cd}} w_{e_c t}. \tag{2}$$

## 4. Evaluation

We evaluate the impact of WSD in our natural language processing pipeline for event-based stock price prediction in two application scenarios. The setup of these evaluations is detailed in Section 4.1. The results are discussed in Section 4.2.

### *4.1. Experimental Setup*

In our evaluation, we focus on the 48 largest companies listed in the NASDAQ-100. Our training set consists of historical stock prices and news articles for these companies, retrieved from Dow Jones Newswires (Dow Jones, 2011) for January 1, 2010 to December 31, 2010. The training set thus constructed contains 16,685 news articles, i.e., 348 articles on average per company, with a standard deviation of 495. We use a similar test set, which covers the historical stock prices and all 17,589 news articles for our considered companies (366 articles on average per company, with a standard deviation of 598), extracted from Dow Jones Newswires for January 1, 2011 to December 31, 2011. As such, our data set covers a wide range of companies from various industries, thus allowing for more generalizable findings. Moreover, our data set and spans over multiple years, which allows us to mitigate potential seasonal or periodic fluctuations on the stock markets.

We use our training set to determine the empirical impact of each of our considered events on each of our 48 companies of interest, 0, 2, and 4 days after an event occurrence. On our test set, we use these trained company-specific weights, i.e., C0, C2, and C4, as well as weights averaged over all companies, i.e., A0, A2, and A4, to predict stock price changes in 0, 2, or 4 days, respectively.

For each parameter set, we evaluate the performance of three distinct stock price prediction methods. Our baseline method, i.e., Baseline, predicts random stock price changes, uniformly distributed between $-1$ and 1. The second prediction method, i.e., Events is our event-driven method as described in Section 3, but without WSD. The third prediction method in our evaluation, i.e., WSDEvents, is the event-driven method as described in Section 3, with the Sinha & Mihalcea (2007) WSD approach enabled. We evaluate the statistical significance of observed differences in performance between these methods by means of a paired two-sample two-tailed t-test.

We evaluate the three stock price prediction methods in two distinct application scenarios. In our first scenario, we use our predicted stock price movements to generate buy (positive predicted change) and sell (negative predicted change) signals, and evaluate the precision of these signals based on the actual historical stock price movements. In the second scenario, we evaluate the excess returns realized by following up on our generated buy and sell signals, compared to the actual $-1.80\%$ returns on the NASDAQ index in 2011. We start with a fictive investment capital of $1,000 for each of our 48 considered companies, and follow up on each generated buy or sell signal by going long (buy) or short (sell) with the full investment capital available for that company at that point in time. We then evaluate the excess returns at the end of this simulated year.

We consider a set of 240 distinct events, stemming from our previous work (Micu et al., 2008). All events are listed in Table 1. When we do not apply WSD, we can detect 23,729 occurrences of these events in our training set (i.e., on average 494 events per company, with a standard deviation of 685), and 26,322 events in our test set (i.e., on average 548 events per company, with a standard deviation of 882). When we apply WSD, the number of detected event occurrences reduces to 16,267 events in our training set (i.e., on average 339 events per company, with a standard deviation of

| Events | | | | |
|---|---|---|---|---|
| share advance | market part up | under expectation | lose suit | new contract |
| share ascend | market percentage grow | under expected value | monopoly* | partnership* |
| share ascent | market percentage rise | widespread loss | suit* | income decline |
| share boost | market percentage up | block deal | product defect | income decrease |
| share climb | market portion grow | delay deal | product flaw | income dip |
| share grow | market portion rise | lose contract | security flaw | income down |
| share heighten | market portion up | lost contract | security problem | income drop |
| share increase | solidify position | miss contract | software defect | income fell |
| share jump | strengthen position | postpone deal | software flaw | income loss |
| share lift | acquire contract | refuse bid | bad economy | income reduce |
| share propel | acquire deal | refuse contract | global crisis | income sink |
| share rise | attain contract | refuse deal | recession* | income slice |
| share up | business deal | reject bid | income boost | income trim |
| stock advance | close deal | reject contract | income climb | earning decline |
| stock ascend | finalize deal | reject deal | income gain | earning decrease |
| stock ascent | gain contract | turn down bid | income growth | earning dip |
| stock boost | gain deal | turn down contract | income increase | earning down |
| stock climb | get contract | turn down deal | income jump | earning drop |
| stock grow | get deal | share decline | income raise | earning fell |
| stock heighten | new deal | share decrease | income rise | earning loss |
| stock increase | sign agreement | share descend | earning boost | earning sink |
| stock jump | sign contract | share dip | earning climb | earning slice |
| stock lift | sign deal | share down | earning gain | earning trim |
| stock propel | anticipate earning | share drop | earning growth | revenue bad |
| stock rise | anticipate profit | share fall | earning increase | revenue decline |
| stock up | exceed expectation | share fell | earning jump | revenue decrease |
| bring out | exceed expected value | share sink | earning rise | revenue dip |
| issue* | expect earning | share slice | revenue boost | revenue down |
| launch* | expect profit | stock decline | revenue climb | revenue drop |
| new application | forecast earning | stock decrease | revenue earn | revenue fell |
| new product | forecast profit | stock descend | revenue gain | revenue loss |
| new service | surpass expectation | stock dip | revenue growth | revenue reduce |
| new software | surpass expected value | stock down | revenue increase | revenue sink |
| new technology | anticipate losings | stock drop | revenue jump | revenue trim |
| publish* | anticipate loss | stock fall | revenue raise | profit bad |
| release* | anticipate losses | stock fell | revenue rise | profit decline |
| dividend boost | bad result | stock sink | profit boost | profit decrease |
| dividend climb | below expectation | stock slice | profit climb | profit dip |
| dividend gain | below expected value | dividend decrease | profit gain | profit down |
| dividend growth | company losings | loss dividend | profit growth | profit drop |
| dividend increase | company loss | low earning | profit increase | profit fell |
| dividend jump | company losses | low income | profit jump | profit loss |
| dividend raise | expect losings | low profit | profit raise | profit reduce |
| dividend rise | expect loss | low revenue | profit rise | profit sink |
| market gain | expect losses | anti trust | turn profit | profit trim |
| market increase | forecast losings | case* | good result | copyright infringement |
| market part grow | forecast loss | lawsuit | alliance | copyright violation |
| market part rise | forecast losses | lose case | expand* | trademark violation |

Table 1: Overview of the lemmas for our considered events. Events marked with * require WSD.

438) and 17,943 events in our test set (i.e., on average 374 events per company, with a standard deviation of 545). Clearly, the application of WSD causes fewer events to be detected, since the WSD process implies that only specific rather than all senses of a lemma trigger the detection of an event. This should, however, mainly reduce the false positive rate for event detection, and thus improve the performance of StockWatcher 2.0.

| Parameters | Baseline | Events | WSDEvents |
|---|---|---|---|
| C0 | 0.491 | 0.502 | 0.506 |
| C2 | 0.486 | 0.500 | 0.500 |
| C4 | **0.494** | 0.505 | 0.493 |
| A0 | 0.491 | 0.509 | 0.509 |
| A2 | 0.486 | **0.530**[a] | **0.533**[a] |
| A4 | **0.494** | 0.503 | 0.497 |

Table 2: Overall precision of buy and sell signals generated by our considered stock price prediction methods. Values marked with [a] differ statistically significantly from the associated values for the Baseline method, at $p < 0.05$. The best performance is printed in bold for each method.

### *4.2. Experimental Results*

We evaluate the stock price movements predicted by our baseline, our event-driven method without WSD, and our event-driven method with WSD, in terms of their performance in two scenarios. The first scenario involves generating buy and sell signals (Section 4.2.1) and evaluating their precision. The second scenario involves guiding a stock investment strategy by these signals (Section 4.2.2) and evaluating the resulting excess returns. We discuss the implications of our findings in Section 4.2.3.

#### *4.2.1. Buy and Sell Signals*

Table 2 shows that including WSD in the event-based prediction process yields negligible differences in signal precision compared to not including WSD in this process. Furthermore, either form of event-based stock price prediction yields modest improvements of, on average, about 4% over our baseline in terms of signal precision. A notable exception to this pattern is the significant 9% precision improvement over the baseline for signals stemming from event-based predictions of stock prices for 2 days after a set of observed events, when using the impact of these events averaged over all companies.

Interestingly, using average rather than per-company weights for events tends to result in signal precision scores that are approximately 3% higher. Furthermore, the precision of buy and sell signals guided by event-based predictions of stock price movements tends to be maximized when evaluating the impact of detected events 2 days after their occurrence.

| Parameters | BASELINE | EVENTS | WSDEVENTS |
|---|---|---|---|
| C0 | **1.873%** | 1.908% [a] | 3.995% [ab] |
| C2 | 1.521% | -2.376% [a] | 2.952% [ab] |
| C4 | -2.795% | 2.869% [a] | 1.829% [ab] |
| A0 | **1.873%** | 4.691% [a] | 5.143% [a] |
| A2 | 1.521% | **6.897%**[a] | **6.215%**[ab] |
| A4 | -2.795% | 2.978% [a] | 6.028% [ab] |

Table 3: Overall excess returns generated by acting upon buy and sell signals generated by our considered stock price prediction methods. Values marked with [a] differ statistically significantly from the associated values for the BASELINE method, and those marked with [b] differ statistically significantly from the associated values for the EVENTS method, at $p < 0.05$. The best performance is printed in bold for each method.

#### *4.2.2. Excess Returns*

The comparably modest improvements in signal precision tend to yield vast, statistically significant improvements in terms of excess returns that are generated by a stock trading strategy that is guided by these signals. Table 3 shows that this is the case when comparing event-based stock price prediction methods to our baseline, as well as when comparing WSD-enabled event-based stock price prediction with a similar method that does not use WSD. On average, event-based stock price movement prediction without WSD already yields excess returns that are significantly higher than those generated by our baseline with over 100%. Using WSD in event-based stock price movement prediction even leads to significant increases in excess returns of, on average, about 200% compared to our baseline, and about 70% compared to not using WSD.

Another interesting observation that can be made in Table 3 is that using average rather than per-company weights for events tends to yield excess returns that are on average about 150% higher. Furthermore, the excess returns generated by a stock trading strategy guided event-based predictions of stock price movements tends to be maximized when evaluating the impact of detected events 2 days after their occurrence.

#### *4.2.3. Discussion*

One of the most striking observations in Sections 4.2.1 and 4.2.2 is that very modest and mostly statistically insignificant changes in the precision of generated buy and sell signals lead to vast improvements in excess returns. An explanation for this phenomenon lies in the earlier observation that, when generating buy and sell signals based on event-based predictions of stock price movements, the number of detected events and thus the number of generated buy and sell signals tends to greatly reduce when WSD is enabled. As a result, the returns of our stock trading strategy are less affected by the noise introduced by ambiguous pieces of text that may or may not be describing relevant events. Moreover, these ambiguous events are typically associated with historical impacts that are between 100% and 400% higher than those found for the non-ambiguous events, thus rendering the ambiguous events typically comparably high-impact.

Another interesting observation for both signal precision and excess returns of a trading strategy guided by event-based stock price movement predictions is that average rather than per-company weights for events tend to yield better results. A possible cause for this lies in the company-specific event weights being more prone to overfitting, and the average weights across all companies being more appropriate for generalizing the impact of specific types of events on businesses in a market. Moreover, this suggests that the impact of specific events does not differ all that much across companies, and instead is rather universal.

The results discussed in Sections 4.2.1 and 4.2.2 also show that performance in terms of signal precision and excess returns tends to be optimized when predicting stock prices about 2 days into the future. This suggests that the full impact of events may not be immediately reflected in stock prices – at least not in the most consistent way. As such, stock prices 2 days after an event reflect this event's impact most reliably.

## 5. Conclusions

In this paper, we have proposed and evaluated an advanced natural language processing pipeline for event-based stock price prediction, that allows for a crucial WSD step to be incorporated in the process of identifying events that can affect stock prices. Our approach is to first identify events in natural language news messages and to subsequently weight these events for their historical impact on stock prices.

The impact of an event on stock prices tends to generalize well across the NASDAQ-100 companies in our validation data. Additionally, this impact can most accurately be predicted in 2 days after an event's occurrence, as demonstrated by the maximized precision in generated buy and sell signals as well as the maximized excess returns when employing a 2-day horizon for the underlying stock price predictions.

When predicting stock price movements based on events, the number of detected events tends to greatly reduce when graph-based WSD using a degree centrality measure is applied in the event detection process. Enabling WSD thus reduces the noise introduced into the stock price movement predictions by comparably high-impact ambiguous events. As a result, modest improvements in the precision of buy and sell signals generated based on these predictions tend to lead to vast improvements in the associated excess returns.

Our encouraging findings warrant various directions of future research. First, a systematic evaluation of various (alternative) types of WSD methods could be an interesting avenue for future research – our considered graph-based WSD approach has the advantages of being unsupervised and producing high-quality output, yet approaches that, e.g., sacrifice output quality for lower computational complexity could be of interest in the financial domain as well. Furthermore, the merits of WSD in event-based stock price prediction could be evaluated in more complex trading strategies, for instance by accounting for dividend rates, borrowing fees, stop-loss orders, hedging, and not going all-in per se when acting upon a generated buy or sell signal. In light of the latter, one could consider modeling and accounting for the confidence in the stock price movement prediction underlying the generated signal. Last, when predicting stock price movements, future research could incorporate a more explicit notion of human sentiment with respect to relevant news articles, detected by means of state-of-the-art automated sentiment analysis techniques.

## References

ACL-SIGLEX (1998). Senseval: Evaluation Exercises for the Semantic Analysis of Text . Available online, `http://www.senseval.org`.

Agirre, E., & Martinez, D. (2000). Exploring Automatic Word Sense Disambiguation with Decision Lists and the Web. In *18th International Conference on Computational Linguistics (COLING 2000)* (pp. 11–19). Association for Computational Linguistics.

Ahmad, K., de Oliveira, P., Manomaisupat, P., Casey, M., & Taskaya, T. (2002). Description of Events: An Analysis of Keywords and Indexical Names. In *3rd International Conference on Language Resources and Evaluation*

*(LREC 2002) Workshop on Event Modelling for Multilingual Document Linking* (pp. 29–35). European Language Ressources Association.

Amancio, D., Oliveira, O., & da F. Costa, L. (2012). Unveiling the Relationship between Complex Networks Metrics and Word Senses. *Europhysics Letters*, *98*, 18002.

Ariyo, A., Adewumi, A., & Ayo, C. (2014). Stock Price Prediction Using the ARIMA Model. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation* (pp. 106–112). IEEE.

Black, E. (1988). An Experiment in Computational Discrimination of English Word Senses. *IBM Journal of Research and Development*, *32*, 185–194.

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter Mood Predicts the Stock Market. *Journal of Computational Science*, *2*, 1–8.

Bonacich, B. (1972). Factoring and Weighing Approaches to Status Scores and Clique Identification. *Journal of Mathematical Sociology*, *2*, 113–120.

Brin, S., & Page, L. (1998). The Anatomy of a Large-scale Hypertextual Web Search Engine. In *7th International Conference on World Wide Web (WWW 1998)* (pp. 107–117). Elsevier.

Brown, P., Della Pietra, S., Della Pietra, V., & Mercer, R. (1991). Word-Sense Disambiguation Using Statistical Methods. In *29th Annual Meeting of the Association for Computational Linguistics (ACL 1991)* (pp. 264–270). Association for Computational Linguistics.

Bruce, R., & Wiebe, J. (1998). Decomposable Modeling in Natural Language Processing. *Computational Linguistics*, *25*, 195–207.

Buscaldi, D., Rosso, P., Pla, F., Segarra, E., & Arnal, E. (2006). Verb Sense Disambiguation Using Support Vector Machines: Impact of WordNet-Extracted Features. In *7th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2006)* (pp. 192–195). Springer volume 3878 of *Lecture Notes in Computer Science*.

Cesarano, C., Dorr, B., Picariello, A., Reforgiato, D., Sagoff, A., & Subrahmanian, V. (2006). OASYS: An Opinion Analysis System. In *AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs (CAAW 2006)* (pp. 21–26). Association for the Advancement of Artificial Intelligence.

Chan, W. (2003). Stock Price Reaction to News and No-News: Drift and Reversal After Headlines. *Journal of Financial Economics*, *70*, 223–260.

Chang, C., Kayed, M., Girgis, R., & Shaalan, K. (2006). A Survey of Web Information Extraction Systems. *IEEE Transactions on Knowledge and Data Engineering*, *18*, 1411–1428.

Chung, Y., Kang, S., Moon, K., & Lee, J. (2002). Word Sense Disambiguation in a Korean-to-Japanese MT System Using Neural Networks. In *2002 COLING Workshop on Machine Translation in Asia (COLING-MTIA 2002)* (pp. 1–7). Association for Computational Linguistics.

Correa Jr., E., & Amancio, D. (2019). Word Sense Induction using Word Embeddings and Community Detection in Complex Networks. *Physica A: Statistical Mechanics and its Applications*, *523*, 180–190.

Correa Jr., E., Lopes, A., & Amancio, D. (2018). Word Sense Disambiguation: A Complex Network Approach. *Information Sciences*, *442–443*, 103–113.

Cottrell, G. (1989). *A Connectionist Approach to Word Sense Disambiguation*. Morgan Kaufmann.

Cowie, J., Guthrie, J., & Guthrie, L. (1992). Lexical Disambiguation Using Simulated Annealing. In *14th International Conference on Computational Linguistics (COLING 1992)* (pp. 359–365). Association for Computational Linguistics.

de Fortuny, E., de Smedt, T., Martens, D., & Daelemans, W. (2014). Evaluating and Understanding Text-Based Stock Price Prediction Models. *Information Processing & Management*, *50*, 426–441.

Decadt, B., Hoste, V., Daelemans, W., & van den Bosch, A. (2004). GAMBL, Genetic Algorithm Optimization of Memory-Based WSD. In *3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3)* (pp. 108–112). Association for Computational Linguistics.

Dow Jones (2011). Dow Jones Newswires. Available online, `https://www.dowjones.com/products/newswires`.

Edmonds, P., & Cotton, S. (2001). SENSEVAL-2: Overview. In *2nd International Workshop on Evaluating Word Sense Disambiguation Systems (Senseval-2)* (pp. 1–5). Association for Computational Linguistics.

Fawcett, T., & Provost, F. (1999). Activity Monitoring: Noticing Interesting Changes in Behavior. In *5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 1999)* (pp. 53–62). Association for Computing Machinery.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Feuerriegel, S., & Prendinger, H. (2016). News-Based Trading Strategies. *Decision Support Systems*, *90*, 65–74.

Francis, W., & Kucera, H. (1964). The Brown Corpus: A Standard Corpus of Present-Day Edited American English, for use with Digital Computers. Brown University Liguistics Department. Available online, `http://icame.uib.no/brown/`.

Freeman, L. (1979). Centrality in Social Networks: Conceptual Clarification. *Social Networks*, *1*, 215–239.

Fuji, A., Inui, K., Tokunaga, T., & Tanaka, H. (1998). Selective Sampling for Example-Based Word Sense Disambiguation. *Computational Linguistics*, *24*, 573–598.

Hagenau, M., Liebmann, M., & Neumann, D. (2013). Automated News Reading: Stock Price Prediction Based on Financial News Using Context-Capturing Features. *Decision Support Systems*, *55*, 685–697.

Hearst, M. (1991). Noun Homograph Disambiguation Using Local Context in Large Corpora. In *7th Annual Conference of the University of Waterloo Centre for the new OED and Text Research* (pp. 1–22). UW Centre for the New OED and Text Research.

Heerschop, B., Goossen, F., Hogenboom, A., Frasincar, F., Kaymak, U., & de Jong, F. (2011a). Polarity Analysis of Texts using Discourse Structure. In *20th ACM Conference on Information and Knowledge Management (CIKM 2011)* (pp. 1061–1070). Association for Computing Machinery.

Heerschop, B., Hogenboom, A., & Frasincar, F. (2011b). Sentiment Lexicon Creation from Lexical Resources. In *14th International Conference on Business Information Systems (BIS 2011)* (pp. 185–196). Springer volume 87 of *Lecture Notes in Business Information Processing*.

Hirst, G., & St-Onge, D. (1998). Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. In C. Fellbaum (Ed.), *WordNet: An Electronic Lexical Database* chapter 13. (pp. 305–332). MIT Press.

Ho, C., Damien, P., Gu, B., & Konana, P. (2017). The Time-Varying Nature of Social Media Sentiments in Modeling Stock Returns. *Decision Support Systems*, *101*, 69–81.

Hogenboom, A., Frasincar, F., de Jong, F., & Kaymak, U. (2015a). Polarity Classification Using Structure-Based Vector Representations of Text. *Decision Support Systems*, *74*, 46–56.

Hogenboom, A., Frasincar, F., de Jong, F., & Kaymak, U. (2015b). Using Rhetorical Structure in Sentiment Analysis. *Communications of the ACM*, *58*, 69–77.

Hogenboom, A., Heerschop, B., Frasincar, F., Kaymak, U., & de Jong, F. (2014). Multi-Lingual Support for Lexicon-Based Sentiment Analysis Guided by Semantics. *Decision Support Systems*, *66*, 43–53.

Hogenboom, A., Hogenboom, F., Frasincar, F., Schouten, K., & van der Meer, O. (2013). Semantics-Based Information Extraction for Detecting Economic Events. *Multimedia Tools and Applications*, *64*, 27–52.

Hogenboom, F., Frasincar, F., Kaymak, U., de Jong, F., & Caron, E. (2016). A Survey of Event Extraction Methods from Text for Decision Support Systems. *Decision Support Systems*, *85*, 12–22.

Hoste, V., Hendrickx, I., Daelemans, W., & van den Bosch, A. (2002). Parameter Optimization for Machine Learning of Word Sense Disambiguation. *Natural Language Engineering*, *8*, 311–325.

Hutchins, W., & Somers, H. (1992). *An Introduction to Machine Translation*. Academic Press.

Iacobacci, I., Pilehvar, M., & Navigli, R. (2016). Embeddings for Word Sense Disambiguation: An Evaluation Study. In *54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)* (pp. 897–907). Association for Computational Linguistics.

Ide, N., & Veronis, J. (1998). Introduction to the Special Issue on Word Sense Disambiguation: the State of the Art. *Computational Linguistics*, *24*, 2–40.

Jansen, B., Zhang, M., Sobel, K., & Chowdury, A. (2009). Twitter Power: Tweets as Electronic Word of Mouth. *Journal of the American Society for Information Science and Technology*, *60*, 2169–2188.

Ji, H. (2010). One Sense per Context Cluster: Improving Word Sense Disambiguation Using Web-Scale Phrase Clustering. In *4th International Universal Communication Symposium (IUCS 2010)* (pp. 181–184). IEEE.

Jiang, J., & Conrath, D. (1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *International Conference Research on Computational Linguistics (ROCLING X)* (pp. 19–33). National Taiwan University.

Joshi, M., Pedersen, T., & Maclin, R. (2005). A Comparative Study of Support Vector Machines Applied to the Supervised Word Sense Disambiguation Problem in the Medical Domain. In *2nd Indian International Conference on Artificial Intelligence (IICAI-05)* (pp. 3449–3468). IICAI.

Kara, Y., Boyacioglu, M., & Baykan, A. (2011). Predicting Direction of Stock Price Index Movement Using Artificial

Neural Networks and Support Vector Machines: The Sample of the Istanbul Stock Exchange. *Expert Systems with Applications*, *38*, 5311–5319.

Kelly, E., & Stone, P. (1975). *Computer Recognition of English Word Senses*. Elsevier.

Kilgarriff, A. (1998). Senseval: An Exercise in Evaluating Word Sense Disambiguation Programs. In *1st International Conference on Language Resources and Evaluation (LREC 1998)* (pp. 1255–1258). European Language Ressources Association.

Koppula, N., Padmaja Rani, B., & Rao, K. (2017). Graph Based Word Sense Disambiguation. In *1st International Conference on Computational Intelligence and Informatics (ICCII 2016)* (pp. 665–670). Springer.

Kristoufek, L. (2013). BitCoin Meets Google Trends and Wikipedia: Quantifying the Relationship Between Phenomena of the Internet Era. *Scientific Reports*, *3*.

Krovetz, R., & Croft, W. (1992). Lexical Ambiguity and Information Retrieval. *ACM Transactions on Information Systems*, *10*, 115–141.

Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., & Allan, J. (2000). Language Models for Financial News Recommendation. In *9th International Conference on Information and Knowledge Management (CIKM 2000)* (pp. 389–396). Association for Computing Machinery.

Le, C., & Shimazu, A. (2004). High WSD Accuracy Using Naive Bayesian Classifier with Rich Features. In *18th Pacific Asia Conference on Language, Information, and Computation (PACLIC 2004)* (pp. 105–114). Association for Computing Machinery.

Leacock, C., & Chodorow, M. (1998). Combining Local Context and WordNet Similarity for Word Sense Identification. In C. Fellbaum (Ed.), *WordNet: An Electronic Lexical Database* chapter 11. (pp. 265–283). MIT Press.

Leacock, C., Chodorow, M., & Miller, G. (1998). Using Corpus Statistics and WordNet Relations for Sense Identification. *Computational Linguistics*, *24*, 147–166.

Lee, Y., & Ng, H. (2002). An Empirical Evaluation of Knowledge Sources and Learning Algorithms for Word Sense Disambiguation. In *2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)* (pp. 41–48). Association for Computational Linguistics.

Lee, Y., Ng, H., & Chia, T. (2004). Supervised Word Sense Disambiguation with Support Vector Machines and Multiple Knowledge Sources. In *3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3)* (pp. 137–140). Association for Computational Linguistics.

Lesk, M. (1986). Automatic Sense Disambiguation Using Machine Readable Dictionaries: How To Tell a Pine Cone from an Ice Cream Cone. In *5th Annual International Conference on Systems Documentation (SIGDOC 1986)* (pp. 24–26). Association for Computing Machinery.

Lin, D. (1997). Using syntactic dependency as local context to resolve word sense ambiguity. In *35th Annual Meeting of the Association for Computational Linguistics (ACL 1997)* (pp. 64–71). Association for Computational Linguistics.

Lu, C. (2010). Integrating Independent Component Analysis-Based Denoising Scheme with Neural Network for Stock Price Prediction. *Expert Systems with Applications*, *37*, 7056–7064.

Lu, Z., Liu, T., & Li, S. (2004). Combining Neural Networks and Statistics for Chinese Word Sense Disambiguation. In *3rd SIGHAN Workshop on Chinese Language Processing (SIGHAN 2004)* (pp. 49–56). Association for Computational Linguistics.

Meijer, K., Frasincar, F., & Hogenboom, F. (2014). A Semantic Approach for Extracting Domain Taxonomies from Text. *Decision Support Systems*, *62*, 78–93.

Micu, A., Mast, L., Milea, V., Frasincar, F., & Kaymak, U. (2008). Financial News Analysis Using a Semantic Web Approach. In A. Zilli, E. Damiani, P. Ceravolo, A. Corallo, & G. Elia (Eds.), *Semantic Knowledge Management: an Ontology-based Framework* chapter 15. (pp. 311–328). Idea Group.

Mihalcea, R. (2005). Unsupervised Large-Vocabulary Word Sense Disambiguation with Graph-Based Algorithms for Sequence Data Labeling. In *Human Language Technology and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)* (pp. 411–418). Association for Computational Linguistics.

Mihalcea, R., & Edmonds, P. (Eds.) (2004). *3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3)*. Association for Computational Linguistics.

Mihalcea, R., & Moldovan, D. (2001). A Highly Accurate Bootstrapping Algorithm for Word Sense Disambiguation. *International Journal on Artificial Intelligence Tools*, *10*, 5–21.

Miller, G., Leacock, C., Tengi, R., & Bunker, R. (1993). Language Models for Financial News Recommendation. In *Workshop on Human Language Technology (HLT 1993)* (pp. 303–308). Association for Computational Linguistics.

Mitchell, M., & Mulherin, J. (1994). The Impact of Public Information on the Stock Market. *Journal of Finance*, *49*, 923–950.

Montoyo, A., Martinez-Barco, P., & Balahur, A. (2012). Subjectivity and Sentiment Analysis: An Overview of the Current State of the Area and Envisaged Developments. *Decision Support Systems*, *53*, 675–679.

Mooney, R. (1996). Comparative Experiments on Disambiguating Word Senses: An Illustration of the Role of Bias in Machine Learning. In *1st Conference on Empirical Methods in Natural Language Processing (EMNLP 1996)* (pp. 82–91). Association for Computational Linguistics.

Navigli, R., & Lapata, M. (2007). Graph Connectivity Measures for Unsupervised Word Sense Disambiguation. In *20th International Joint Conference on Artificial Intelligence (IJCAI 2007)* (pp. 1683–1688). Association for the Advancement of Artificial Intelligence.

Navigli, R., & Lapata, M. (2010). An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*, 678–692.

Navigli, R., & Velardi, P. (2005). Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*, 671–674.

Ng, H. (1997a). Exemplar-Based Word Sense Disambiguation: Some Recent Improvements. In *2nd Conference on Empirical Methods in Natural Language Processing (EMNLP 1997)* (pp. 208–213). Association for Computational

Linguistics.

Ng, H. (1997b). Getting Serious about Word Sense Disambiguation. In *ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?* (pp. 1–7). Association for Computational Linguistics.

Niu, C., Li, W., Srihari, R., Li, H., & Crist, L. (2004). Context Clustering for Word Sense Disambiguation Based on Modeling Pairwise Context Similarities. In *3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3)* (pp. 187–190). Association for Computational Linguistics.

Nuij, W., Milea, V., Hogenboom, F., Frasincar, F., & Kaymak, U. (2014). An Automated Framework for Incorporating News into Stock Trading Strategies. *IEEE Transactions on Knowledge and Data Engineering*, *26*, 823–835.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting Stock and Stock Price Index Movement Using Trend Deterministic Data Preparation and Machine Learning Techniques. *Expert Systems with Applications*, *42*, 259–268.

Patwardhan, S., Banerjee, S., & Pedersen, T. (2003). Using Measures of Semantic Relatedness for Word Sense Disambiguation. In *4th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2003)* (pp. 241–257). Springer volume 10762 of *Theoretical Computer Science and General Issues*.

Pedersen, T. (2001). A Decision Tree of Bigrams is an Accurate Predictor of Word Sense. In *2nd Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies (NAACL 2001)* (pp. 1–8). Association for Computational Linguistics.

Pedersen, T., & Bruce, R. (1997). Distinguishing Word Senses in Untagged Text. In *2nd Conference on Empirical Methods in Natural Language Processing (EMNLP 1997)* (pp. 197–207). Association for Computational Linguistics.

Peramunetilleke, D., & Wong, R. (2002). Currency Exchange Rate Forecasting from News Headlines. *Australian Computer Science Communications*, *24*, 131–139.

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and Application of a Metric on Semantic Nets. *IEEE Transactions on Systems, Man and Cybernetics*, *19*, 17–30.

Resnik, P. (1995). Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *14th International Joint Conference on Artificial Intelligence (IJCAI 1995)* (pp. 448–453). Morgan Kaufmann.

Roh, T. (2007). Forecasting the Volatility of Stock Price Index. *Expert Systems with Applications*, *33*, 916–922.

Rosen, R. (2006). Merger Momentum and Investor Sentiment: The Stock Market Reaction to Merger Announcements. *Journal of Business*, *79*, 987–1017.

Rui, H., Liu, Y., & Whinston, A. (2013). Whose and What Chatter Matters? The Effect of Tweets on Movie Sales. *Decision Support Systems*, *55*, 863–870.

Sabidussi, G. (1966). The Centrality Index of a Graph. *Psychometrika*, *31*, 581–603.

Sanderson, M. (1994). Word Sense Disambiguation and Information Retrieval. In *17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1994)* (pp. 142–151). Association for Computing Machinery.

Schumaker, R., Zhang, Y., Huang, C., & Chen, H. (2012). Evaluating Sentiment in Financial News Articles. *Decision*

*Support Systems*, *53*, 458–464.

Seo, Y., Giampapa, J., & Sycara, K. (2002). *Text Classification for Intelligent Agent Portfolio Management*. Technical Report CMU-RI-TR-02-14 Carnegie Mellon University. Available online, `http://ri.cmu.edu/pub_files/pub4/seo_young_woo_2002_1/seo_young_woo_2002_1.pdf`.

Sinha, R., & Mihalcea, R. (2007). Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity. In *1st IEEE International Conference on Semantic Computing (ICSC 2007)* (pp. 363–369). IEEE.

Tsatsaronis, G., Vazirgiannis, M., & Androutsopoulos, I. (2007). Word Sense Disambiguation with Spreading Activation Networks Generated from Thesauri. In *20th International Joint Conference on Artificial Intelligence (IJCAI 2007)* (pp. 1725–1730). Association for the Advancement of Artificial Intelligence.

van Bommel, J. (2003). Rumors. *Journal of Finance*, *58*, 1499–1520.

van de Camp, M., & van den Bosch, A. (2012). The Socialist Network. *Decision Support Systems*, *53*, 761–769.

Veronis, J., & Ide, N. (1990). Word Sense Disambiguation with Very Large Neural Networks Extracted from Machine Readable Dictionaries. In *13th International Conference on Computational Linguistics (COLING 1990)* (pp. 389–394). Association for Computational Linguistics.

Walker, D. (1987). Machine Translation: Theoretical and Methodological Issues. In S. Nirenburg (Ed.), *Knowledge Resource Tools for Accessing Large Text Files* chapter 14. (pp. 247–261). Cambridge University Press.

Wu, Z., & Palmer, M. (1994). Verb Semantics and Lexical Selection. In *32nd Annual Meeting of the Association for Computational Linguistics (ACL 1994)* (pp. 133–138). Association for Computational Linguistics.

Yarowsky, D. (1992). Word-Sense Disambiguation using Statistical Models of Roget's Categories Trained on Large Corpora. In *14th International Conference on Computational Linguistics (COLING 1992)* (pp. 454–460). Association for Computational Linguistics.

Yarowsky, D. (1994). Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French. In *32nd Annual Meeting of the Association for Computational Linguistics (ACL 1994)* (pp. 88–95). Association for Computational Linguistics.

Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *33rd Annual Meeting of the Association for Computational Linguistics (ACL 1995)* (pp. 189–196). Association for Computational Linguistics.

Yarowsky, D. (2000). Hierarchical Decision Lists for Word Sense Disambiguation. *Computers and the Humanities*, *34*, 179–186.

Yu, X., Liu, Y., Huang, X., & An, A. (2012). Mining Online Reviews for Predicting Sales Performance: A Case Study in the Movie Domain. *IEEE Transactions on Knowledge and Data Engineering*, *24*, 720–734.

Yu, Y., Duan, W., & Cao, Q. (2013). The Impact of Social and Conventional Media on Firm Equity Value: A Sentiment Analysis Approach. *Decision Support Systems*, *55*, 919–926.

Zhang, X. (2006). Information Uncertainty and Stock Returns. *Journal of Finance*, *61*, 105–137.

Zhu, D., Lappas, T., & Zhang, J. (2018). Unsupervised Tip-Mining from Customer Reviews. *Decision Support Systems*, *107*, 116–124.