Enhancing Sentiment Analysis of Financial News by Detecting Negation Scopes

Nicolas Prölochs  
University of Freiburg  
Freiburg, Germany  
nicolas.proellochs@is.uni-freiburg.de

Stefan Feuerriegel  
University of Freiburg  
Freiburg, Germany  
stefan.feuerriegel@is.uni-freiburg.de

Dirk Neumann  
University of Freiburg  
Freiburg, Germany  
dirk.neumann@is.uni-freiburg.de

Abstract

Sentiment analysis refers to the extraction of the polarity of source materials, such as financial news. However, measuring positive tone requires the correct classification of sentences that are negated, i.e. the negation scopes. For example, around 4.74% of all sentences in German ad hoc announcements contain negations. To predict the corresponding negation scope, related literature commonly utilizes two approaches, namely, rule-based algorithms and machine learning. Nevertheless, a thorough comparison is missing, especially for the sentiment analysis of financial news. To close this gap, this paper uses German ad hoc announcements as a common example of financial news in order to pursue a two-sided evaluation. First, we compare the predictive performance using a manually-labeled dataset. Second, we examine how detecting negation scopes can improve the accuracy of sentiment analysis. In this instance, rule-based algorithms produce superior results, resulting in an improvement of up to 9.80% in the correlation between news sentiment and stock market returns.

1. Introduction

Human readers usually recognize a sentence as negated intuitively or by applying grammatical rules. Such negating phrases occur frequently in financial news. For example, Figure 1 shows that as many as 4.74% of all sentences in regulated ad hoc announcements of German firms contain one negating phrase, while 0.48% contain even two or more, possibly nested, negations. In fact, identifying the negated parts of a statement is especially crucial when analyzing the sentiment of financial news, since sentiment analysis is highly vulnerable to negations; any of the corresponding sentences are likely to be classified erroneously. Consequently, handling negations is an inevitable prerequisite [1]–[3] in order to measure positive tone. However, previous research points out “that positive tone is difficult to accurately measure, since positive words are easily negated in ways difficult to programmatically identify” [2]. To overcome this gap, this paper proposes and evaluates algorithms with the aim of predicting negated parts of sentences.

Negations can appear in various different forms in financial news. In fact, simple negations can invert not only the meaning of single words, but also of whole phrases. Consequently, parts of a sentence whose meaning is inverted are called negation scope. For example, the meaning of grown in “the economy has not grown” is inverted by the word not. Furthermore, negations can also invert the meanings of sentences implicitly; e.g. “the company has invented a new product, it was the first and last time”.

In order to detect negation scopes, previous research has proposed various approaches that can primarily be...
Table 2. Comparison of algorithms for predicting negation scopes.

<table>
<thead>
<tr>
<th>Rule-Based Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Straightforward to implement</td>
</tr>
<tr>
<td>+ Low computational costs</td>
</tr>
<tr>
<td>- Implicit negations problematic</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Machine Learning: Hidden Markov Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Can adjust to domain-specific features/language</td>
</tr>
<tr>
<td>+ Many degrees of freedom in model</td>
</tr>
<tr>
<td>- Computationally expensive</td>
</tr>
<tr>
<td>- Supervised learning requires training data</td>
</tr>
</tbody>
</table>

grouped (e.g. [4]) into two categories: (a) *rule-based approaches* and (b) algorithms for both supervised and unsupervised *machine learning*. Both methods have their advantages, as well as their limitations (see Table 2). For example, rule-based algorithms struggle to discover implicit linguistic negations, not to mention peculiarities, such as sarcasm or irony. However, rule-based methods are straightforward to implement, whereas a computationally more expensive alternative is provided by machine learning in the form of Hidden Markov models. These models use labeled data (if possible) in order to replicate actual human negation detection.

The purpose of this paper is to improve the detection of negation scopes in financial news. As a main contribution, this paper explores a clear and intriguing application – the sentiment analysis of financial news. This allows us to investigate and compare predicted negation scopes in practice. In contrast to previous research, which typically focuses on either rule-based approaches or machine learning methods, we aim to implement both. The linguistic rules are based on different predefined window sizes and/or make use of word classes to replicate grammatical rules. In the case of machine learning, we implement different variants of Hidden Markov models using both supervised and unsupervised learning, in order to predict the negated parts of a sentence. Finally, this paper evaluates these predictions in terms of predictive performance and computational time using two dimensions: we examine the accuracy of a manually-labeled dataset consisting of negated sentences from financial news. Since the chosen labels are biased by a subjective interpretation, the stock market reaction of investors serves as an objective measure. Thus, we compute sentiment values for financial news and compare the correlation of these sentiment values with the corresponding stock market returns. Altogether, this two-pronged evaluation reveals a path on how to improve accuracy when analyzing financial news sentiment.

The remainder of this paper is structured as follows. Section 2 provides an overview of related literature which similarly aims at negation scope prediction. In Section 3, we explain our methodology for the evaluation of financial news and illustrate the way in which we implement negation rules and suitable Hidden Markov models for our task. Afterwards, all aforementioned methods are compared (Section 4) using a two-pronged approach: first, we measure the predictive performance on a manually-labeled dataset and, second, evaluate how the identification of negation scopes helps to improve sentiment analysis.

2. Related Work: Negation Scope Detection

This section provides an overview of previous publications that also study how sentiment analysis can be improved by negation detection. This basically affects any context [5]–[7] using sentiment analysis but is particularly true when it comes to financial news [1]–[3]. Managers can easily negate words in ways that are difficult to identify computationally [2], [3]. Investigations about 10-K reports indicate that companies often frame negative news using positive words, and rarely relay positive news in terms of negative words [1]. A thorough survey on the role of negation in sentiment analysis, as well as how to embed negation scopes, is found in [5].

When using blog entries as a corpus, negation scope detection based on heuristic rules results in significant improvements [8]. A similar approach inverts the meaning only within windows of a fixed size [9], which is evaluated using IMDb reviews. Efforts in opinion mining systems on several manually labeled corpora lead to robust cross-domain performance of rule-based approaches [10]. Research on the same data using similar approaches points out problems with implicit negations and recommends the application of machine learning approaches [11]. In the domain of machine learning, authors propose the use of conditional random fields [6] to predict negation scopes for the sentiment analysis of product reviews. Finally, we note that negation scope detection is also a frequent research topic (e.g. [4], [12]) in information retrieval aimed at medical reports.

Hence, this paper addresses the following research question: we propose and compare algorithms to predict negation scopes. Thus, we utilize linguistic rules and machine learning. To our best knowledge, this is
the first study that predicts negation scopes in order to enhance the sentiment analysis of financial news.

3. Methodology

This section introduces our research methodology. In a first step, each announcement is subject to pre-processing steps which transform the running text into machine-readable tokens. Next, we present the underlying methods in order to determine negation scopes using both rule-based algorithms and Hidden Markov models.

3.1. Data Preprocessing

Before performing the actual sentiment analysis, several operations are involved in a preprocessing phase. The individual steps are as follows:

- **Cleaning.** By using a list of cut off patterns, we omit contact addresses and formatting in order to extract only the textual components of ad hoc Announcements.

- **Tokenization.** Each announcement is split into sentences and single words named tokens [13].

- **Stop word removal.** Words without a deeper meaning, such as the, is, of, … are named stop words [14] and can thus be removed. We use a list of 571 stop words [15].

- **Part-of-Speech Tagging.** Part-of-speech (POS) tagging is the process of annotating words, based on definition and context, with the corresponding word class, i.e. part of speech. Here, we use the so-called Brill tagger [16], which returns a total of 36 parts of speech. In a subsequent step, we map these parts of speech to a reduced subset, consisting of a total of 11 word classes. This reduced set is given in Table 3, where we also manually define a word class negations, consisting of our chosen negation words (cf. Table 7).

- **Stemming.** In computational linguistics, stemming refers to the process that reduces inflected words to their stem [14]. One usually aims to map related words to the same stem, even if this stem is not itself a valid root form, as long as inflected forms are grouped together. Thus, we annotate words with their stems by using the so-called Porter stemming algorithm [17].

Table 3. Parts of speech (i.e. word classes), with additional group for negations, used in both corpus and Hidden Markov models.

<table>
<thead>
<tr>
<th>Adjectives</th>
<th>Nouns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adverbs</td>
<td>Numerals</td>
</tr>
<tr>
<td>Articles</td>
<td>Prepositions</td>
</tr>
<tr>
<td>Conjunctions</td>
<td>Pronouns</td>
</tr>
<tr>
<td>Interjections</td>
<td>Verbs</td>
</tr>
<tr>
<td>Negations</td>
<td></td>
</tr>
</tbody>
</table>

3.2. Rule-Based Detection of Negation Scopes

In order to predict the scope of negations accurately, a common approach relies upon linguistic rules which try to imitate the grammatical structure of a sentence. More precisely, these negation rules seek for the presence of negation words and invert the meaning of certain surrounding parts. A simple example is given by

“the sales could not be increased”.

In this sentence, the originally positive word increased is inverted by the preceding negation phrase not; increased must be now be regarded in a negative context. However, the effect of negation words can vary strongly. For instance, the first word following a negation phrase might be negated, whereas the meaning of all succeeding words remains the same, such as in

“the production has not grown, but we are hopeful for the future”.

Here, the negation word not indicates that the meaning of the main clause is inverted, while it does not affect the subordinate clause.

What the above examples all have in common is the fact that readers apply linguistic rules to resolve negations scopes in order to understand the meaning. Consequently, we utilize negation rules as a benchmark to predict negation scopes. Individual negation rules are as follows:

**Benchmark: No Inverting**

**Rule 1** Negation words have no impact on the negation scope.

**Negation of Sentence Fragments**

**Rule 2** Invert all subsequent words within a sentence after the occurrence of a negation word [9].

**Rule 3** Negation of the whole sentence, if one or more negation words occur [9].

**Rule 4** Improve Rule 3 in order to handle double negatives. If an odd number of negation words occur in a sentence, the whole sentence is considered as negated. With an even number of negation words, the whole sentence is considered as positive.
Negation of Fixed Window Sizes

Rule 5 Invert a given window of words following a negation word, where the window size is set to 1–5 words [9], [11].

Rule 6 Improve Rule 5 in order to handle double negatives within the window. Only if an odd number of negation words occur within the window will the entire window be considered as negated.

Negation using Part-of-Speech Patterns

Rule 7 Invert all words succeeding a negation word until a certain part of speech is reached [8].

Rule 8 Invert a sequence of words before and after a negation word until a certain part of speech is reached. For example, the negation window starts at the last article before the negation word and ends with the next noun. The maximum window size is set to 4 words before the negation word and 3 words after the negation word.

Rule 9 Same as Rule 8, but invert only adjectives.

Negation by Modeling Objects

Rule 10 This rule allows to invert the subsequent object. Thus when encountering a negation word, all immediately following pronouns, prepositions and articles are inverted. Then, all immediately following adjectives and adverbs are inverted. Finally, all subsequent nouns are negated. This process stops prematurely if the part of speech pattern is interrupted earlier. However, the minimum window size is set to 3. Overall, this is supposed to model most linguistic structures of an object [18].

3.3. HMM-Based Detection of Negation Scopes

Rule-based approaches involve certain limitations, for example, these methods cannot adapt to domain-specific features or particularities of the chosen prose. To overcome these restrictions, we also implement algorithms from statistical learning. Out of all the possible strategies for machine learning, the most common choice in order to predict negation scopes is the so-called Hidden Markov model, which this section describes and adapts to our problem.

When true states of a model are not directly observable but the possible effects are, such a system can be described by a Hidden Markov model, as specified according to the following definition [19].

Definition (Hidden Markov Model). A Hidden Markov model is a generative, probabilistic model which consists of $N$ not directly observable, hidden states

$$S = \{S_1, S_2, \ldots, S_N\}$$

and $M$ distinct observation symbols per state, i.e. the emission alphabet, given by

$$V = \{v_1, v_2, \ldots, v_M\}.$$ 

Let $q^{(t)}$ denote the current state at time $t$. A Hidden Markov model further contains a state transition probability distribution $A = \{a_{ij}\}$, given by

$$a_{ij} = P\left[q^{(t+1)} = S_j \mid q^{(t)} = S_i\right] > 0$$

for $1 \leq i \leq N$, $1 \leq j \leq N$.

Additionally, an observation symbol probability distribution $B = \{b_{jk}\}$ in state $S_j$ is given by

$$b_{jk} = P\left[v_k \mid q^{(t)} = S_j \text{ and } v_k = O_i\right]$$

for $1 \leq j \leq N$, $1 \leq k \leq M$, and an initial distribution $\pi = \{\pi_i\}$ with

$$\pi_i = P\left[q^{(1)} = S_i\right]$$

for $1 \leq i \leq N$.

With all the above parameters, the Hidden Markov model can be rewritten as a 3-tuple

$$\lambda = (A, B, \pi).$$

Consequently, a Hidden Markov model resembles a double stochastic process, where after each period $t$, the system can remain in the current state or move to another state. From each state $S = \{S_1, S_2, \ldots, S_N\}$, the system can reach another state in a single step, enforced by $a_{ij} > 0$ in Eq. (3). As a visual example, Figure 4 shows a stochastic process with two states $S = \{S_1, S_2\}$. In order to integrate a second stochastic process, each of these states can now emit a symbol $O_i$ of the emission alphabet $V$ at time $t$. Then, the model generates a sequence of observable states

$$O(T) = [O_1, O_2, \ldots, O_T]$$

until a time step $T$.

In machine learning literature [19], [20], Hidden Markov models are linked with the three following tasks.

Evaluation. Computing the probability of an observation sequence $O(T) = [O_1, O_2, \ldots, O_T]$ for a model $\lambda = (A, B, \pi)$ with e.g. the Forward algorithm.

1. Other types of Hidden Markov models also allow $a_{ij} \geq 0$. 

962
Decoding. Computing the corresponding state sequence $Q = [q^{(1)}, q^{(2)}, \ldots, q^{(T)}]$ to an observation $O_T$ and a model $\lambda$ using the so-called Viterbi algorithm.

Learning. Adjustment of the parameters $\lambda = (A, B, \pi)$ to maximize $P(O(T) \mid \lambda)$ via a method named Baum-Welch algorithm.

We proceed as follows in order to specify Hidden Markov models to predict negation scopes. The directly observable states are given by

$$S = \{\text{Negated}, \neg \text{Negated}\}. \tag{8}$$

Each of these states emits at step $t$ a value $v$ from the emission alphabet $V$. Here, we compare the predictive performance across three variants of the emission alphabet: first, we set the emission alphabet to a list of part-of-speech tags. This setup is illustrated in Figure 5. Second, we choose the actual words as emission symbols – giving a considerably larger set but allowing for higher accuracy. Third, we choose only the word stems as a comparison. In a next step, we determine both the transition probabilities and emission probabilities. This is achieved by employing either the Viterbi algorithm for supervised learning (using a manually annotated dataset) or by implementing the Baum-Welch algorithm for unsupervised learning. All variants of Hidden Markov models are compared in terms of predictive performance for negation scope detection in the following sections.

3.4. Sentiment Analysis

In order to calculate the polarity of news announcements, we utilize the frequently-employed approach of the so-called Net-Optimism sentiment measure [21], [22]. This approach assumes that a news announcement with a positive message correlates with more positive words and vice versa. Thus, the Net-Optimism measure rates the content according to the frequencies of words in pre-defined dictionaries classified as either positive or negative. Let $P$ denote the positive word list and $N$ the negative word list. Then, a news announcement $A = [w_1, \ldots, w_n]$, represented by a vector consisting of words $w_i$, has a Net-Optimism measure of $S_{NO}(A) \in [-1, +1]$, which is defined by

$$S_{NO}(A) = \frac{|\{i \mid w_i \in P\}| - |\{i \mid w_i \in N\}|}{|A|}. \tag{9}$$

Consequently, the sentiment variable $S_{NO}(A)$ measures the difference between positive and negative words normalized by the total number of words. Although many other approaches can be found in the literature [23]–[25], Net-Optimism provides not only robust results [26], but its simplicity makes later comparisons straightforward.

4. Evaluation

This section describes our datasets, as well as the experimental setup. Using the methods from the previous section, we proceed by comparing the aforementioned strategies to predict negation scopes, namely, both rule-based algorithms and Hidden Markov models. The contribution of our evaluation (see Figure 6) is twofold. On the one hand, Section 4.2 compares the predictive performance on a manually-labeled dataset to yield a direct and comparative measure. On the other hand, negation scope detection is likely to be combined with various tasks in natural language processing (NLP), most probably joint with sentiment analysis. When it
Evaluation

Validation: Labeled Dataset

NLP Application: Sentiment Analysis

+ Computation Time

Figure 6. Two-sided evaluation compares predictive performance, as well as computation time, using both a manually-labeled dataset and sentiment analysis as a natural language processing (NLP) application.

comes to applications like sentiment analysis, it is not always the case that the model that performs best on a dataset manually labeled by humans also provides the best results in an application scenario. Consequently, Section 4.3 further evaluates algorithms for negation scope detection within sentiment analysis.

4.1. Dataset

Our news corpus originates from the German regulated ad hoc announcements\(^2\) between January 2004 to June 2011. As a requirement, each announcement must have at least 50 words and be written in English. Our final corpus consists of 14,463 ad hoc announcements. In research, ad hoc announcements are a frequent choice [24] when it comes to evaluating and comparing methods for sentiment analysis. Additionally, this type of news corpus shows several advantages: ad hoc announcements must be authorized by company executives, the content is quality-checked by the Federal Financial Supervisory Authority\(^3\) and several publications analyze their relevance to the stock market – finding a direct relationship (e.g. [27]). To study the stock market reaction, we use the daily stock market returns of the corresponding company, originating from Thomson Reuters Datastream. For those firms where we were not able to retrieve stock market returns, we use the corresponding ad hoc announcements only in the manually-labeled dataset. In order to measure the sentiment of these announcements, we utilize the Loughran and McDonald Financial Sentiment Dictionary [1], [28]. The dictionary contains 354 entries with

\(^2\) Kindly provided by Deutsche Gesellschaft für Adhoc Publizität (DGAP).

\(^3\) Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin).


<table>
<thead>
<tr>
<th>Word</th>
<th>Frequency in Sentences in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>not</td>
<td>60.65</td>
</tr>
<tr>
<td>hardly</td>
<td>28.29</td>
</tr>
<tr>
<td>*n’t</td>
<td>6.98</td>
</tr>
<tr>
<td>rather</td>
<td>2.96</td>
</tr>
<tr>
<td>denied/denies</td>
<td>0.61</td>
</tr>
<tr>
<td>without</td>
<td>0.23</td>
</tr>
<tr>
<td>never</td>
<td>0.15</td>
</tr>
<tr>
<td>denied</td>
<td>0.13</td>
</tr>
</tbody>
</table>

† Included additionally

Figure 8. Relative frequency of negation words from Table 7 that appear in news corpus.

4.2. Comparison using the Manually-Labeled Dataset

We start our evaluation with a manually-labeled dataset consisting of 400 extracted sentences. For this purpose, we compare the forecast performance in negation scope prediction of the rules from Section 3.2 with the performance of several Hidden Markov model implementations from Section 3.3. In this context, we use 10-fold cross-validation [20] to calculate accuracy, precision, recall and \(F_1\) score of the forecast. The
results for different negation rules are listed in the second block of Table 9.

Negations affect the meaning only of small sentence fragments, as the meaning of words at the beginning of a sentence are unlikely to be inverted. Thus, data labels are unevenly distributed and the accuracy defining the Gold standard goes well beyond 50%. In fact, our benchmark model (which does not recognize negations at all) reveals an accuracy of 79.56 %, equivalent to the share of non-negated words in the dataset. Consistent with [11], rule-based algorithms reveal the highest performance using a negation window with a fixed size of 3 words. Apparently, we achieve better overall results when handling nested negation. Furthermore, negation scopes by modeling objects give similar results with an $F_1$ score of 0.7297 and an accuracy of 89.62 %. Our Hidden Markov model implementations produce inferior results. In the case of supervised learning, Hidden Markov models show the highest $F_1$ score when using words stems as training data. Unsupervised learning using the Baum-Welch algorithm leads to, generally speaking, inferior performance results.

4.3. Comparison using Sentiment Analysis as a NLP Application

Now, we evaluate how negations in financial news affect sentiment analysis. In a first step, we calculate a sentiment value using the Net-Optimism measure for each announcement depending on the rule-based algorithms from Section 3.2 and the Hidden Markov model implementations from Section 3.3. In a second step, we analyze the correlation between the sentiment values and the corresponding daily stock market returns. The results of the correlation values are listed in the third block of Table 9.

A benchmark model, in which occurring negations are left untreated, gives a correlation of 0.0798 between sentiment value and stock market return. Similar to [11], we achieve the highest correlation when using a negation window with a fixed size of 5 words. Several other rules reveal superior results compared to the benchmark case, for example, those rules that model the object with the help of part-of-speech patterns (e.g. using a negation window until the next conjunction or noun). All of the aforementioned rules achieve an improvement in correlation between 9.71 % and 9.80 %. Evidently, the best performing rule for the manually-labeled dataset negates a fixed window of 3 words, while 5 words are more beneficial for sentiment analysis.

In contrast, Hidden Markov models for negation scope detection show inferior results to the benchmark model. Out of all the considered machine learning approaches, we find the highest correlation when training a Hidden Markov model using the (unmodified) words – improving the correlation by 5.14 % in comparison to the benchmark model.

Interestingly, the impact of negations in sentiment analysis seems to be more significant than for the predictive performance of the manually-labeled dataset. This is an effect which can be explained by the following cause: when measuring sentiment in textual materials, only those words count that also appear in the dictionary. These dictionary entries apparently occur more often after a negation word and thus are more dependent on correct negation classification. Overall, 4.74 % of all sentences contain a negation according to Table 7. Each of these sentences contain an average of 0.906 dictionary entries, whereas this number drops to 0.596 for sentences without negation. When examining the position of dictionary entries in negated sentences more closely, 39.56 % precede a negation, while 60.46 % occur subsequently.

4.4. Comparison of Computation Times

Ultimately, we consider the necessary computational resources in the identification of negation scopes by comparing average computation times. All codes are implemented in C# along with the .NET framework 4 and using the Accord.NET framework for the Hidden Markov models. The mean computation times for different rules in milliseconds per sentence are listed in the last column of Table 9. In this comparison, we exclude the time needed for training the models. All timings are measured on an Intel Xeon CPU E7-8850 2.00 GHz with 64 GB RAM (per core) and 64-bit Windows Server 2008 R2 (SP 1) running with 12 GFlops per core.

Rules that negate either fixed window sizes or sentence fragments require a computation time of approximately 0.0030 ms. Furthermore, rules using part-of-speech patterns rely on the corresponding POS tags for each word in a sentence. To compute these tags, the algorithms require, on average, a computation time that is 7.1 times longer.

When it comes to machine learning approaches, they are likely to exceed the runtimes of rule-based algorithms. For example, the Viterbi algorithm, used for decoding the state sequences in our Hidden Markov model implementations, requires significantly more computational operations. A Hidden Markov model using part-of-speech tags as emission symbols requires a 7.6 times longer computation time than the benchmark model. A HMM using words stems requires 428.1
Table 9. Negation scope detection by rule-based and machine learning approaches compared across labeled dataset and sentiment analysis.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Benchmark: No Inverting</th>
<th>Evaluation: Labeled Dataset</th>
<th>Evaluation: Sentiment Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Recall</td>
<td>Precision</td>
</tr>
<tr>
<td>1</td>
<td>No negation recognition</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Negation of all following words</td>
<td>0.7814</td>
<td>0.2851</td>
</tr>
<tr>
<td>3</td>
<td>Negation of the whole sentence</td>
<td>1</td>
<td>0.2044</td>
</tr>
<tr>
<td>4</td>
<td>Whole sentence incl. double negations</td>
<td>0.7327</td>
<td>0.1808</td>
</tr>
<tr>
<td>5a</td>
<td>Negation of fixed window of 1 word</td>
<td>0.1917</td>
<td>0.9232</td>
</tr>
<tr>
<td>5b</td>
<td>Negation of fixed window of 2 words</td>
<td>0.3552</td>
<td>0.8647</td>
</tr>
<tr>
<td>5c</td>
<td>Negation of fixed window of 3 words</td>
<td>0.4750</td>
<td>0.7866</td>
</tr>
<tr>
<td>5d</td>
<td>Negation of fixed window of 4 words</td>
<td>0.5606</td>
<td>0.7149</td>
</tr>
<tr>
<td>5e</td>
<td>Negation of fixed window of 5 words</td>
<td>0.6134</td>
<td>0.6445</td>
</tr>
<tr>
<td>6a</td>
<td>Nested negations in window of 1 word</td>
<td>0.3793</td>
<td>0.9054</td>
</tr>
<tr>
<td>6b</td>
<td>Nested negations in window of 2 words</td>
<td>0.5419</td>
<td>0.8731</td>
</tr>
<tr>
<td>6c</td>
<td>Nested negations in window of 3 words</td>
<td>0.6548</td>
<td>0.8133</td>
</tr>
<tr>
<td>6d</td>
<td>Nested negations in window of 4 words</td>
<td>0.7218</td>
<td>0.7530</td>
</tr>
<tr>
<td>6e</td>
<td>Nested negations in window of 5 words</td>
<td>0.7436</td>
<td>0.6879</td>
</tr>
<tr>
<td>7a</td>
<td>Negation until next article</td>
<td>0.6585</td>
<td>0.6222</td>
</tr>
<tr>
<td>7b</td>
<td>Negation until next verb</td>
<td>0.4185</td>
<td>0.5072</td>
</tr>
<tr>
<td>7c</td>
<td>Negation until next noun</td>
<td>0.5433</td>
<td>0.7937</td>
</tr>
<tr>
<td>8a</td>
<td>Last article until next noun</td>
<td>0.6184</td>
<td>0.5004</td>
</tr>
<tr>
<td>8b</td>
<td>Last verb until next noun</td>
<td>0.6216</td>
<td>0.6064</td>
</tr>
<tr>
<td>8c</td>
<td>Last conjunction until next noun</td>
<td>0.6280</td>
<td>0.5075</td>
</tr>
<tr>
<td>8d</td>
<td>Last article until next adjective</td>
<td>0.6480</td>
<td>0.4981</td>
</tr>
<tr>
<td>9</td>
<td>Negation of only adjectives</td>
<td>0.0437</td>
<td>0.6857</td>
</tr>
<tr>
<td>10</td>
<td>Negation until next object</td>
<td>0.8962</td>
<td>0.7802</td>
</tr>
</tbody>
</table>

**Detection using Hidden Markov Models**

- HMM using POS for supervised learning | 0.1790 | 0.6199 | 0.2777 | 0.8097 | 0.0825 | 3.36 | 0.0220 |
- HMM using words for unsupervised learning | 0.3260 | 0.7089 | 0.4467 | 0.8349 | 0.0827 | 3.70 | 2.0076 |
- HMM using stems for supervised learning | 0.3816 | 0.7230 | 0.4996 | 0.8437 | 0.0839 | 5.14 | 1.2490 |
- HMM using POS for unsupervised learning | 0.2413 | 0.2583 | 0.2929 | 0.3582 | 0.0775 | −2.81 | 0.0219 |
- HMM using words for unsupervised learning | 0.2413 | 0.2583 | 0.2495 | 0.7032 | 0.0815 | 2.18 | 1.9608 |
- HMM using stems for unsupervised learning | 0.2823 | 0.2498 | 0.2651 | 0.6800 | 0.0812 | 1.75 | 1.2497 |
times and, on top of that, a HMM using words requires an approximately 688.0 times longer computation time. Finally, it is noteworthy that the unsupervised learning methods for Hidden Markov models require an especially huge additional prior computation time to build and train the models.

5. Conclusion

Negations are a fundamental stylistic device in written texts with the objective of inverting the meaning of both words and sentences. For instance, German ad hoc announcements, as an example of financial news, contain negations in as many as 4.74% of all sentences. Such a non-marginal proportion of negations can pose a substantial obstacle when performing a sentiment analysis. According to [2], “the results for positive words are mixed because many times, negative phrases are wrapped in positive words”. To alleviate such a possible cause of noise, this paper evaluates the different approaches for negation scope detection. We follow a two-sided approach. We evaluate the predictive performance not only on a manually-labeled dataset, but examine the benefits of predicted negation scopes in an application from natural language processing: more precisely, how sentiment analysis of financial news can be improved. With an increasing prevalence of the sentiment analysis of financial news in both research and practice, further effort will inevitably focus on improving individual tools for sentiment analysis, such as handling negation scopes.

As its main contribution, this paper compares methods to predict negation scopes. Rule-based algorithms lead to superior results in both applications stated. These reveal a forecast accuracy of up to 89.87% on a manually-labeled dataset. In sentiment analysis, we achieve an improvement in correlation between sentiment value and stock market return of 9.80% in comparison to a benchmark model with no handling of negations. Interestingly, the impact of negations in sentiment analysis seems to be more significant than for the predictive performance of the manually-labeled dataset. Words at the beginning of a sentence, which are unlikely to be negated words, contribute to an accurate forecasting of the manually-labeled dataset. In sentiment analysis, only the words in the dictionary are crucial for the performance. These words presumably occur more often after a negation word and thus are more strongly influenced by correct negation classification.

In future work, we will advance the above methods in three directions. First, our evaluation would benefit greatly from the comparison of varying subsets of negation terms. Second, it might be beneficial to include additional news sources from other domains to test the robustness. In this case, commodities, initial public offerings or real estate markets are intriguing examples. Third, extending the selection of machine learning models could possibly open an avenue for further gains in accuracy. For example, conditional random fields are related to Hidden Markov models, but use an undirected graphical model that relaxes certain assumptions on both input and output distributions. In addition, reinforcement learning holds potential as an alternative which can learn negation patterns in order to directly maximize the correlation between sentiment analysis and stock market returns, as measured by a reward function. In fact, we are not aware of any publications that utilize reinforcement learning to directly improve the accuracy of sentiment analysis.

References


