Improving Sentiment Analysis with Document-Level Semantic Relationships from Rhetoric Discourse Structures

Joscha Märkle-Huß
University of Freiburg
Freiburg, Germany
mjoscha@gmail.com

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

Helmut Prendinger
National Institute of Informatics
Tokyo, Japan
helmut@nii.ac.jp

***Abstract*—Conventional sentiment analysis usually neglects semantic information between (sub-)clauses, as it merely implements so-called bag-of-words approaches, where the sentiment of individual words is aggregated independently of the document structure. Instead, we advance sentiment analysis by the use of rhetoric structure theory (RST), which provides a hierarchical representation of texts at document level. For this purpose, texts are split into elementary discourse units (EDU). These EDUs span a hierarchical structure in the form of a binary tree, where the branches are labeled according to their semantic discourse. Accordingly, this paper proposes a novel combination of weighting and grid search to aggregate sentiment scores from the RST tree, as well as feature engineering for machine learning. We apply our algorithms to the especially hard task of predicting stock returns subsequent to financial disclosures. As a result, machine learning improves the balanced accuracy by 8.6 percent compared to the baseline.**

1. Introduction

Sentiment analysis refers to methods and applications of computational linguistics that identify and extract the subjective tone in textual materials. Among the most common methods are bag-of-words models that consider the frequency of words and/or their N-gram combinations [1]. However, these approaches usually rely exclusively on the number of occurrences of certain relevant words or phrases. As such, these methods are not capable of taking into account the actual semantic relationships between parts of the document, individual sentences or even subclauses [2], [3]. Accordingly, these methods often struggle to achieve a favorable performance for longer documents and, hence, new approaches are desired [4].

To overcome the previous limitations, this paper advances sentiment analysis by incorporating additional features that represent the semantic structure. We thus utilize rhetoric discourse structure (RST) to create a discourse tree and then propose two innovative approaches to leverage the information therein. To our knowledge, previous research leaves many questions on this topic unanswered.

Research has proposed multiple approaches – besides manual annotation – to extract discourse information in an automated fashion. For instance, [5] uses syntax and lexical information to obtain sentence- but not document-level semantic structures. Similarly, [6] studies the internal composition of sentences by considering the syntax tree structure with the help of recursive deep neural networks. Other approaches utilize low-level discourse-based features in the form of connectors for (sub-)clauses [7]–[9]. However, these are limited to sentences where such a connector is present and thus cannot reflect the discourse of a full document. Furthermore, examples of a document-level discourse analysis include shift-reduce discourse parsers to create RST annotations [10] or discriminative frameworks for rhetorical analysis [11].

Previous work has started to utilize semantic information to label opinions without the objective of analyzing sentiment. For instance, [12] develops a shallow semantic representation and then assigns custom feature structures to a set of 4 groups of top-level opinion categories. However, this approach only serves the purpose of classifying opinions, while it does not address how to extract a sentiment score for individual texts. A similar work improves the automated detection of opinion frames, but a demonstration of how to use this information for sentiment analysis is missing [13].

In fact, previous research has largely neglected to enrich sentiment analysis through semantic information. Among the few works that do, is a study that utilizes sentence-level RST, as well as a topicality measure, in order to assign different sentiment weights to sentences depending on their relevance [14]. However, the authors do not adjust for the different types of argumentative relationships between clauses. This is complemented by [15], which uses a Bayesian model to evaluate the sentiment of hotel reviews using sentiment, aspect and discourse information.

Closest to our research is an approach by [3] that incorporates the rhetorical structure at document level, but this paper neither presents a method of how to combine all discourse units in an optimal fashion nor compares methodological variants, including machine learning approaches. Many machine learning algorithms struggle with this type of problem as it is virtually impossible to encode with a fixed-length vector while preserving its order and context [4].

URI: http://hdl.handle.net/10125/41288
ISBN: 978-0-9981331-0-2

The only investigation of machine learning for rhetoric-structure-based sentiment analysis is given in [2], which utilizes recursive neural networks (RNN). As such, it struggles with small datasets, as in this case, where we need to rely on hand-crafted feature engineering. Furthermore, RNNs provide hardly any descriptive insights for inferring the importance of different semantic relationships.

As our primary contribution, this paper proposes and compares different methods for improving sentiment analysis with semantic information. We utilize rhetoric structure theory to disassemble documents into individual elementary discourse units (EDUs), which are organized in the form of binary trees [16]. Hence, one can compute the sentiment for a whole document by aggregating the sentiment values from individual EDUs, i. e. the leaves of the tree. By following this approach, one can take knowledge of the rhetoric discourse structure into account. We then evaluate and compare two methods that are based on (1) a weighting scheme tuned by a grid search and (2) a novel feature engineering approach to support machine learning. In the following evaluation, we demonstrate our methods for discourse-based sentiment analysis with the help of financial disclosures. We utilize the insights to explain the meaning and importance of semantic information for applications of computational linguistics.

The rest of this paper is structured as follows. Section 2 presents how rhetoric structure theory uncovers the semantic discourse of a document. This discourse structure serves as an input to our novel methods for sentiment analysis in Section 3. We then evaluate how the semantic structure can improve sentiment analysis in Section 4. Section 6 finally provides both a discussion and a conclusion.

2. Rhetoric Structure Theory

Rhetoric structure theory (RST) is a classification approach to computationally determine the organization of narrative materials. As such, it aims at identifying the coherence or structure of a text by evaluating the relationships between different (sub-)clauses.

The smallest information item is called an elementary discourse unit (EDU), which forms the smallest, indivisible segment of sentences [16]. Each EDU is put into a hierarchical relationship to the other EDUs, depending on the rhetoric structure of the document [11]. This hierarchy is ultimately mapped to a binary tree to simplify the computational processing.

An example of such a tree is shown in Figure 1. Here, EDUs are assigned only to the leaves (i. e. N_{21}, S_{22}, S_{35}, N_{36} and N_{24}, while all other internal nodes are required to represent the hierarchical relations of these EDUs. Following this, RST assigns one of 18 different relationship types (as listed in Table 5) to each pair of neighboring child nodes. These relationship types provide more detailed information on the coherence linking the EDUs.

Furthermore, every node is classified as either a so-called nucleus or a satellite. The former, nucleus N_{ij}, contains the core information, whereas satellite S_{ij} provides supporting

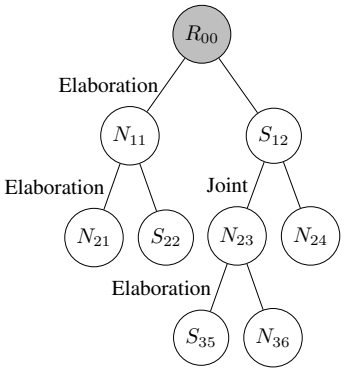


Figure 1. Schematic rhetoric discourse structure as binary tree with root R_{00}, nuclei N_{ij} and satellites S_{ij} at depth i and number j.

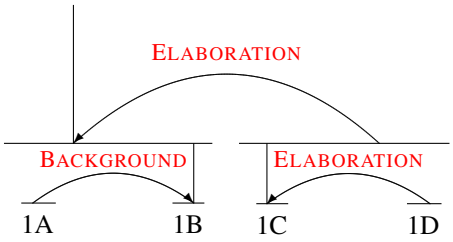
information. Here, the index i denotes the depth, while j enumerates all potential nodes at depth i from left to right. An example is visualized in Figure 1, where exemplary relation types, as well as nucleus and satellite information, are shown. The root node R_{00} is not classified.

Previous research has proposed various methods – statistical approaches, machine learning and rule-based algorithms – to automate the parsing of RST trees. For example, [17] presents a rule-based discourse parser, which has been advanced into a probabilistic, sentence-level parser named SPADE [5]. Furthermore, the CODA project[1] aims at dialogue generation and, as part of this, annotates texts with RST information. A detailed overview of approaches and methods for manual and automated parsing is provided in [11]. In addition, the so-called HILDA parser[2] [18] employs support vector machine classification to perform the identification of EDUs and label their type of relationship. As labeling is combined with a greedy bottom-up tree building approach, high accuracies can be achieved while a linear time complexity is ensured.

This research utilizes HILDA for several reasons [19]: its engine relies on a classification based on a support vector machine to obtain RST features at document level. In contrast to SPADE, it can thus create document-level RST trees. CODA has a stronger focus on the segments and thus only generates EDUs without internal nodes representing the tree structure, whereas HILDA captures the full RST tree across all levels – from root to leaves. This is confirmed by a comparison in [10] where HILDA achieves the highest performance in identifying the hierarchy of texts earning an F1-score of 83.0 (only human annotation outperforms this with an F1-score of 88.7). In terms of nuclearity analysis, e. g., the identification of nuclei and satellites, HILDA also achieves a favorable F1-score of 68.4, which is slightly below that of newer state-of-the-art algorithms which achieve F1-scores of up to 71.1. Only recently have algorithms such as discourse parsing from linear projections succeeded in outperforming HILDA at the task of relationship identifi-

[1] COherent Dialogue Automatically Generated from Text, see http://computing.open.ac.uk/coda/.

[2] HIgh-Level Discourse Analyzer.



[After adjusting for exchange rates ,]^{1A} [the Company generated an increase in sales of about two percent.]^{1B} [At EUR 21.2 million , provisional net profit has turned out less than the figure]^{1C} [budgeted at the beginning of 2003.]^{1D}

Figure 2. Classification of two sample sentences from an an exemplary press release by HILDA. Arrows point from satellites towards nuclei.

cation, achieving an F1-score of up to 61.75 compared to HILDA's F1-score of 54.8 [10].

Figure 2 shows the markup created by HILDA for two sample sentences from a press release in our financial news corpus. In this figure, arrows always point towards the nucleus. Accordingly, the first sentence contains 1A and 1B, while also representing the first nucleus. The second sentence (1C and 1D) appears to be an elaboration, as well as a satellite to the first. When looking at the leaves, we observe that the first sentence is split into two parts: 1B forms the nucleus, whereas the satellite 1A provides a background to 1B.

3. Novel Methods for Sentiment Analysis with Discourse Structure

The previous literature review shows that knowledge is scarce on how to leverage RST information in order to improve the performance of sentiment analysis. Publications in this field of study largely fail to explain the meaning and importance of the various pieces of semantic information.

Fig. 3 shows our proposed methodology for analyzing the sentiment of natural language using semantic relationships. Accordingly, Section 3.1 describes our corpus in the form of regulated financial news, before we describe the steps of pre-processing and sentiment scoring in Section 3.2. These steps thus obtain sentiment values for each EDU. Subsequently, we aggregate these granular sentiment scores at document level. For this purpose, we utilize (1) a simple weighting rule, which we optimize via a grid search (see Section 3.3). The grid search also provides insights into the importance of different parts of texts for readers. In addition, we propose (2) a machine learning approach, whereby we can measure variable importance based on random forests. Section 3.4 then illustrates how we use feature engineering to apply random forests to our dataset and we introduce random forests in Section 3.5.

3.1. Corpus

Methods for sentiment application are often evaluated using datasets that are common in research, such as movie reviews. We specifically refer in the following to the even harder challenge of predicting stock market movements following financial disclosures. These disclosures are mainly targeted at investors, who can then incorporate the disclosures into their decision-making. Research shows that investors do not only react to the facts in texts, but also soft information in written communication such as the perceived sentiment of a message [20]–[24]. The corresponding evidence points out that investors in financial markets frequently refer to textual information in order to decide upon exercising ownership in stocks [24]–[28]. Such a relationship becomes especially evident in the link between the content of ad hoc announcements and the subsequent stock market response [29], [30].

Our corpus consists of 12,932 ad hoc announcements[3] published between January 2004 and June 2011 in the English language. This data forms a popular choice in academic research [29]–[33]. We take the first 80 % of the announcements in time for training and the remaining 20 % for testing. This splitting in adherence to the timing prevents from training on the description of an event and then applying it to the pre-event time span. For instance, one wants to avoid training a classifier with data from after a financial crisis and then map it to the time beforehand in order to detect this crisis. Subsequently, we link each announcement to the nominal stock market return of the publication day. In addition, we use HILDA to dissect all announcements into a total of 491,833 EDUs. Figure 4 reports the number of EDUs appearing across different depths of the RST hierarchy. Interestingly, the depth of RST trees can reach as many as 95 levels, as the trees are of binary structure and thus easily grow very deep.

Table 5 shows the different types of contextual relationships connecting individual EDUs as detected by HILDA, as well as their absolute and relative frequency in our corpus. As a reference, we also compare their frequencies to the typical distribution of relationships types in the rightmost column based on the CODA parallel corpus [34]. Evidently, the most common relationship types in financial disclosures are elaborations and joints. This is especially interesting as the occurrences of these relation types by far exceed the share given in the CODA parallel corpus. A possible reason might be the rather technical language used in financial disclosures. It is thus likely that authors intentionally avoid reader-engaging or complex text structures, such as contrasts and conditions, in order to make the texts comprehensible. Other types of relationships are not present in the corpus under study (such as evaluation, topic-change and topic-comment), possibly also because of deficits of HILDA to identify some of them [18].

[3]Kindly provided by Deutsche Gesellschaft für Ad-Hoc-Publizität (DGAP).

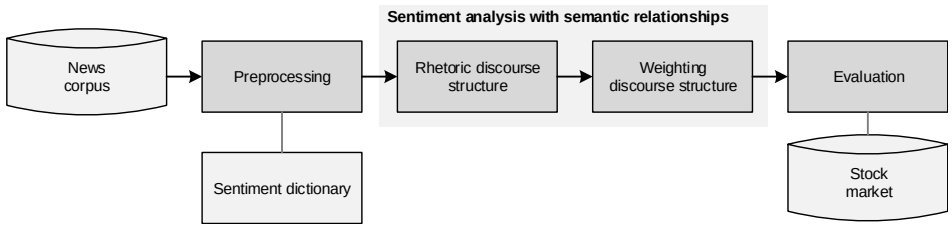


Figure 3. Flow diagram visualizes the concept of our proposed methodology for sentiment analysis using sematic relationships.

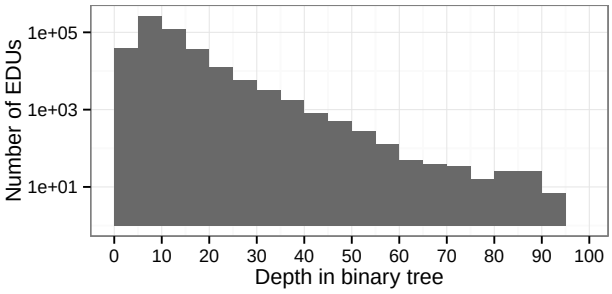


Figure 4. Plot shows the depths of all elementary discourse units (EDUs) in the dataset with a logarithmic y-axis.

TABLE 5. COMPARISON OF RELATIONSHIP TYPES IN THE FINANCIAL DISCLOSURE CORPUS COMPARED WITH CODA PARALLEL CORPUS.

Relation	Financial disclosures		CODA parallel corpus
	Count	Percentage	Percentage
Elaboration	311,808	65.11%	21.31%
Joint	104,717	21.87%	4.12%
Same-Unit	20,825	4.35%	0.00%
Background	9,875	2.06%	1.03%
Attribution	8,740	1.83%	4.12%
Textual-Organization	7,031	1.47%	0.00%
Comparison	4,545	0.95%	1.72%
Temporal	2,650	0.55%	0.69%
Enablement	1,964	0.41%	0.00%
Contrast	1,822	0.38%	20.27%
Summary	1,656	0.35%	1.37%
Condition	1,443	0.30%	8.25%
Manner-Means	874	0.18%	2.75%
Cause	676	0.14%	1.37%
Explanation	275	0.06%	18.90%
Evaluation	0	0.00%	8.59%
Topic-Change	0	0.00%	0.00%
Topic-Comment	0	0.00%	4.81%
Total	**478,901**	**100.00%**	**100.00%**

3.2. EDU sentiment calculation

We perform the following preprocessing steps, which are common in text mining [35]:

1) **Data filtering.** First, we select only texts written in English and that are longer than 150 words. We further exclude so-called penny stocks, i. e. we remove all announcements for which the stock valuation is below 5 monetary units.
2) **Content extraction.** In addition, we remove all parts of the content not belonging to the main message. This includes disclaimers and contact information, but also hyperlinks and HTML tags.
3) **RST parsing.** We then use HILDA to parse the announcements and obtain the corresponding RST tree. This step also splits every document into the individual EDUs.
4) **Content cleaning.** We remove punctuations, numbers and stop words. In this regard, stop words are those which do not contain actual content, such as *the*, *is* and *of*. We use a list from previous research containing 571 stop words [36]. We subsequently convert all characters to lower case.
5) **Stemming.** We perform stemming, whereby all inflected word forms are mapped onto their root. For this purpose, we use the Porter stemmer [37].

We then proceed to calculate sentiment scores for each EDU. For this purpose, we implement a dictionary-based approach to sentiment analysis, whereby we utilize Henry's finance-specific dictionary [25]. This dictionary provides a list of frequent words in financial communication, which are classified as either positive or negative. For instance, the positive word list contains entries such as *achieve*, *expansion* or *improve*, while the negative one features expressions like *decline*, *penalty* or *weakened*. We then simply count the frequency of positive and negative words and insert them into the ratio formula

$$\sigma_{ij} = \frac{P - N}{T} \tag{1}$$

for each EDU j at depth i, where P, N, and T are the number of positive, negative and total words, respectively.

As an example, let us consider the sentence *"the strong$^+$ demand increase$^+$ exceeded$^+$ our expectations in this challenging$^-$ environment"*. It contains three positive words (*strong*, *increase*, *exceed*) and one negative word (*challenging*). Excluding stop words, the sentence comprises of seven words in total and, thus, we obtain a sentiment score of $\frac{3-1}{7} = \frac{2}{7}$. This approach is frequent in related works and has the advantage of performing well for financial communication and also yields robust results when dealing with fairly extensive documents [21].

3.3. Node weighting with grid search

After computing sentiment scores for each individual EDU, a simple approach would be to average the senti-

ment scores across all EDUs but this would neglect any information from the RST tree. Hence, we utilize the former approach as a baseline and, as a remedy, propose a weighting scheme that allows us to combine these based on the structure of the RST tree. To achieve this goal, we develop a weighting scheme that discriminates between two elementary properties of the RST tree, namely, the node type (nucleus vs. satellite) and the depth of each individual node. The former enables us to encode the assumption that nuclei are of greater importance than satellites, while the latter can penalize sentiment scores from nodes depending on their depth in the RST tree. For instance, we expect nodes in closer proximity to the root to be of higher relevance. Both discriminators are translated into a parameter of the weighting scheme.

Mathematically, we introduce two parameters controlling the weighting. These are a weight ω_{ij} to discriminate between nucleus and satellite, as well as D_{i+1} to penalize EDUs located farther away from the root in the RST tree. We thus yield the sentiment score of a document by combining the weighted sentiment values of all children in a recursive fashion, i. e.

$$\sigma_{ij} = D_{i+1}\big[\sigma_{i+1,2j-1}\,\omega_{i+1,2j-1} + \sigma_{i+1,2j}\,\omega_{i+1,2j}\big] \tag{2}$$

with weights ω_{ij} depending on the node type and weight D_{i+1} depending on the depth.

We now describe the choice of ω_{ij} and D_{i+1}. First, we take the sentiment scores σ_{ij} assigned to the EDUs and then weigh them according to an exponential decay. The weight ω_{ij} for a node i at depth j is given by

$$\omega_{ij} = \begin{cases} \alpha, & \text{for nucleus } N_{ij}, \\ 1-\alpha, & \text{for satellite } S_{ij}. \end{cases} \tag{3}$$

The variable α is later optimized by a grid search. Second, we define

$$D_i = \begin{cases} \frac{1+\beta}{i_{\max}} i - \beta, & \beta < 0, \\ \frac{\beta-1}{i_{\max}} i + 1, & \beta \geq 0, \end{cases} \in \big[\,|\beta|\,, 1\big] \tag{4}$$

with $i_{\max}$ being the maximum depth of the corresponding sub-tree and where $\beta \in [-1, +1]$ is the second parameter of the grid search. If $\beta < 0$, it puts an additional penalty on deeper parts of the RST tree, whereas $\beta > 0$ means that deeper components are considered more important.

The above approach is a generalization of two variants used in the literature. On the one hand, it contains a weighting based on the node type. In prior research, nuclei have received a weight of 1 and satellites a weight of 0 [3]. A different study proposed the use of weights 1.5 and 0.5 for nuclei and satellites, respectively [14]. In contrast, our approach benefits from the advantage that we do not set predefined weighting parameters, but, instead, perform a grid search to directly identify the optimal choice. Furthermore, we include an additional factor for weighting based on the depth in the tree. This thus allows us to compare the importance of nuclei and satellites at different levels in the binary tree.

Finally, we note that we tried different functions to model the depth weight D_i. Our experiments all point towards the use of an exponential decay, since this works well, even when facing very deep RST trees. In fact, we need an algorithm that can easily handle trees of depths between 10 and 90, as the depth of trees varies strongly according to Figure 4. For instance, we tried a linear decay to weigh by depth. However, this results into an inferior predictive performance. This is likely due to the fact that information near the root or the leaves is more relevant than in the middle of the RST tree.

3.4. Feature engineering for machine learning

In addition to the above weighting, we propose an approach for incorporating RST information into machine learning in order to compute sentiment scores at document level. We especially strive to incorporate machine learning as this hopefully allows us to overcome the rather rigid structure of the above weighting scheme. In contrast, many algorithms from machine learning entail a highly flexible component and can thus detect non-linear relationships between variables in order to make more accurate predictions. For this purpose, we now explain how we engineer appropriate features that provide input to a machine learning algorithm of our choice. That is, the RST tree itself cannot be encoded by means of a fixed-length numerical vector and, as a remedy, we propose a way to encode relevant characteristics in such a vector.

Feature engineering describes the selection and encoding of predictors or other data. It is especially important for machine learning algorithms as good feature engineering can improve the algorithm performance [38] and increases the transparency of the underlying process. In our case, feature engineering is required to bring the binary tree into a format supported by machine learning algorithms (such as random forests).

Based on the RST tree, we generate features as follows: we choose a maximum depth i to generate a vector

$$\left[\sigma_N, \sigma_S, \ldots, \sigma_{N\ldots N}, \ldots, \sigma_{S\ldots S}\right]^T. \tag{5}$$

Here, we insert sentiment values σ_{ij} depending on their semantic information, i. e. the corresponding location in the RST tree. Hence, the labels represent a sequence of nuclei and satellites, which one has to follow from the root to locate the desired node. For example, σ_{SN} denotes the sentiment score of the nucleus in the first satellite branch.

This encoding entails several benefits. Foremost, it works well even when only a few observations are present, as it reduces the complexity of the original RST tree significantly. Second, one can derive additional explanatory insights by measuring the variable importance for each node.

As part of our research process, we also explored other forms of encoding, e. g. by using additional features for the combination of node types and their relationships at each hierarchical level. To take information about the relationship into account, we encoded relationship types into our vector

and combined it with nuclearity information. Unfortunately, this resulted in an inferior predictive performance. Evidently, this is due to the limited information contained in relationship types, since a predominant share of relationship types come as eithre elaboration or joint. In contrast to this, hierarchical information and nuclearity convey more decisive information than the relationship type. In future works, one might like to consider the use of a different RST parser than HILDA, since newer parsers might achieve better accuracy in detecting relationship attributes and, therefore, one can potentially improve predictions further [3].

3.5. Random forest

Random forests [39], [40] represent an ensemble learning method that constructs a large collection of de-correlated decision trees during training. The output is then the majority vote over all individual trees. The training algorithm for random forests applies *bagging* (also called bootstrap aggregating) to the single tree learner. Bagging repeatedly selects a total of B bootstrap samples from the training set and fits trees – using the Gini impurity – to these samples. The number of bootstrap samples B is a free parameter. Interestingly, increasing the number of trees often tends to decrease the variance of the model without increasing the bias.

In addition, random forests can be used to rank the importance of variables [39] as follows. First, a random forest is fit to the dataset. For each split in the tree, the prediction performance is (1) recorded for the out-of-bag portion of the data and then (2) the same is computed after permuting each variable. The corresponding difference between them is averaged over all trees and normalized. Finally, the larger this difference, the stronger the influence of that specific variable on the performance of the resulting random forest.

In our analysis, we decided upon random forests for several reasons. First of all, random forests cope well with nonlinearities and provide good protection against overfitting. As a result, they achieve high predictive performance with much effort of tuning. Finally, they allow for a ranking of variables according to their importance. This thus provides explanatory insights and contributes to our understanding of how the reader responds to the discourse in a document.

4. Evaluation

This section compares the performance of both the weighting scheme with grid search and feature engineering with machine learning. Table 6 reports the prediction accuracy for all baselines to the RST-based approaches. We discuss their results in the subsequent sections.

In the following, we compare the predictive performance across different metrics. First of all, we compute the accuracy, which measures the ratio of right guesses. A different variant of it, the so-called balanced accuracy, neutralizes imbalances in the dataset, as it is calculated by taking the mean of sensitivity and specificity. Here, sensitivity describes the share of correctly identified downward movements from the set of actual negative stock returns. Similarly, specificity provides the same measure for upward movements. Finally, the F1-score is another metric for measuring predictive performance. It takes precision and recall into account or, more precisely, it is calculated from the geometric mean of precision and recall. Thus, the best possible F1-score takes the value 1 (or 100 %), while the worst accounts for 0.

4.1. Baselines

Table 6 reports the predictive performance across the baselines. The first baseline is given by the performance of using no predictor, i. e. guessing the mean return from the training set. This baseline already exhibits an above-average accuracy of 59.761 %, which originates from a severe imbalance of positive (54 %) and negative (46 %) stock returns in our dataset. Therefore, we adjust for imbalances between different labels by computing the balanced accuracy. Hence, the balanced accuracy using no predictors amounts to exactly 50 % as expected. The imbalance is also mirrored by the sensitivity and specificity values. As this approach always predicts an upward movement due to the imbalanced dataset, the sensitivity amounts to 0 in this case. The specificity, however, computes to 1, as all upward movements are correctly identified. When using no predictors, unsurprisingly, the F1-score is 0.

We now investigate the second baseline given by the average EDU sentiment. This benchmark results from a dictionary-based sentiment, i. e. the average sentiment across all EDUs, while ignoring all RST-related information. The corresponding results are listed in the second row of Table 6. Here, we observe a slight improvement as manifested by a higher sensitivity. At the same time, this results in a marginal decrease of accuracy due to the large class imbalance in the dataset. However, both the balanced accuracy and the F1-score increase, which indicates that the sentiment values actually contain discriminative power.

The above baselines illustrate the challenge of predicting stock market movements using the sentiment of financial news. In our case, sentiment analysis only entails a small improvement over random guessing in terms of the balanced accuracy. In contrast, sentiment is a considerably better predictor in other datasets, such as movie reviews, where accuracies go as high as 70 % with an approach comparable to ours [3].

4.2. Weighting scheme with grid search

We perform a grid search to optimize the weighting parameters α and β in order to achieve a high predictive performance when computing a document-level sentiment score for each news announcement in the corpus. Here, we use a binomial regression with 10-fold cross-validation in order to map the sentiment scores onto the the stock market return subsequent to a news disclosure. The resulting performance is shown in Figure 7. Maximum accuracy is achieved with a nucleus weighting of $\alpha = 0.4$, while all tested values of

Method	Accuracy	Balanced Accuracy	Sensitivity	Specificity	F1-Score
Baseline					
No predictor	59.761	50.000	0.000	100.000	0.000
Avg. EDU sentiment	59.752	50.014	0.228	99.800	0.456
RST-based approaches					
Best weighting	59.798	50.057	0.160	99.954	0.319
RST-ML	60.490	54.315	83.371	25.260	71.889

TABLE 6. PERFORMANCE (IN %) COMPARED ACROSS DIFFERENT ALGORITHMS.

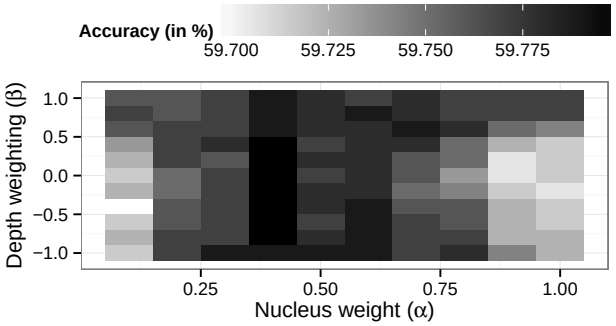


Figure 7. Accuracy from binomial regression when performing a two-dimensional grid search by varying weights for both node type and the depth.

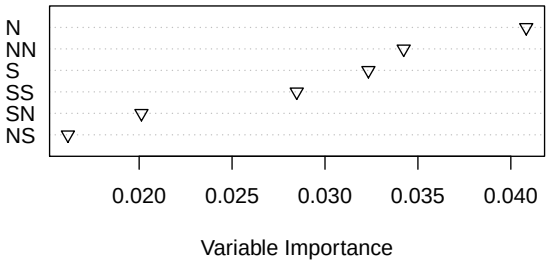


Figure 8. Variable importance of RST information measured across 1,000 trees. The labels correspond to the sequence of nodes encountered when moving from the root to the node with the specific sentiment value (e. g. SN denotes the first satellite branch, followed by a nucleus).

β yield a fairly similar, favorable performance. We observe no clear trend regarding the choice of β for depth-based weighting.

Altogether, RST information is capable of significantly improving the predictive performance. The best result from the grid search features an accuracy of 59.798 % and a balanced accuracy of 50.057 %.

4.3. Machine learning with RST features

The last row in Table 6, named RST-ML, shows the predictive performance based on our extracted features from the RST tree. Here we utilize a random forest with 1,000 trees, as this method from machine learning seamlessly handles non-linearities in an out-of-box fashion. Compared to the average EDU sentiment, we see a relative improvement of 8.6 % in the balanced accuracy, which now totals 54.315 %. Hence, the combination of RST and machine learning performs extraordinarily well for the prediction task under study. This is confirmed by the relatively high F1-score of 71.9 %.

In addition, Figure 8 presents the variable importance, which ranks the predictors by relevance. When predicting the stock market reaction, the most important sentiment information is thus contained in the N and NS branches of the binary tree. This is closely followed by S and we thus deduce that information aggregated at higher levels plays a more important role, while the nuclearity at deeper tree levels is less relevant. This is especially confirmed by SS being the fourth most important variable, even before SN and NS.

5. Discussion and Implications

Even in the face of recent advances in natural language processing (NLP), machine learning still struggles when incorporating semantic or contextual information for text processing. The reason behind this difficulty is of a fundamental nature as it is virtually impossible to encode a text with a fixed-length vector while preserving its order and context [4]. Therefore, research must identify new paths towards analyzing natural language while adapting to the order of words and sentences. On a word level, deep learning approaches this problem formulation with the help of recurrent neural networks. Recurrent neural networks essentially process a sentence (or short text fragment) word-by-word and maintain a short code vector that represents the knowledge and meaning from the first to the current word. However, experiments using ad hoc announcements reveal that text fragments must be short and performance improvements are nevertheless challenging to achieve [42].

While deep learning allows one to include context information when processing a single sentence, this paper proposes an intriguing method for processing the semantic structure within (extensive) documents. Therefore, we parse the rhetoric discourse structure of a given document and thereby unveil its internal semantic skeleton. Thus, we learn

TABLE 9. COMPARISON OF PREDICTIVE APPROACHES WITH DIFFERENT TYPES OF CONTEXT INFORMATION FOR PREDICTING STOCK MARKET RETURNS BASED ON ENGLISH-LANGUAGE AD HOC ANNOUNCEMENTS.

Author	Context level	Approach to context/semantics	Sentiment source	Comparison	Text fragment
Hagenau et al. 2013 [41]	Word groups	n-grams	Chi-based feature selection	Single words retrieved from corpus vs. 2-word combinations	Full content
Feuerriegel and Fehrer 2016 [42]	Word order	Recurrent relationship between words	Implicit feature selection	Random forest vs. recursive neural network	Headlines only
Pröllochs et al. 2015 [43], Pröllochs et al. 2016 [33],	Word-level	Negation scopes	Loughran and McDonald Financial Sentiment Dictionary [26]	No negation recognition vs. negation detection	Full content
This paper	Document structure	Semantic relationship between clauses	Henry Finance Dictionary [25] No feature/word optimization	Average EDU sentiment vs. weighting from RST	Full content

how pieces of information from different sentences interact and which storyline a document follows. Consequently, we contribute to research on text mining by providing a novel method for including semantic information when processing texts. Furthermore, this paper provides a proof of concept that semantic relationships can enhance the accuracy when analyzing the sentiment of textual materials.

In the context of financial markets, researchers use different approaches to improve the prediction of stock market changes based on the tone of financial disclosures. Depending on their experiment setups, they achieve relative improvements of predictive accuracy in the range of 5 % to 15 %. An overview of this research – all using the same ad hoc corpus – is shown in Table 9.

The corresponding improvements achieved by this research, depend heavily on the filter criteria, which makes comparisons intractable. Overall, the relative improvement from our RST information is of similar magnitude to other approaches – but without readjusting the polarity scores of words. In fact, all other approaches in Table 9 implicitly compute new polarity scores for each word or feature. In contrast, we leave the individual scores unchanged and only exploit the semantic structure to achieve our improvements. This demonstrates the largely unused value of semantic information for natural language processing. Moreover, additional improvements are on the horizon, since some of these approaches are likely to enhance the accuracy of sentiment analysis further when combined with features from rhetoric discourse structure.

Ultimately, a better understanding of human language can spark business innovations in multiple areas in the decision-support domain. In the future, embracing semantic relationships is likely to become even more relevant with the current rise of natural language processing. Prevalent examples include voice control for mobile devices, such as Siri and similar voice control services, as a convenient form of human-computer interaction. Such applications need to understand cross references to previous user commands, interpret their relationship and extract their meaning in order to execute the desired action. Semantic information can also improve automated decision-support on the basis of almost any source of textual materials. For example, investors or automated traders can fine-tune their algorithms in order to obtain better investment decisions from financial news. Furthermore, in the case of recommender systems and opinion mining, texts provide decision-support by tracking the public mood to measure brand perception or judge the launch of a new product based on blog posts, comments, reviews or tweets. Altogether, the relevance of our methodology goes beyond these examples and essentially comprises almost all text-based applications of individuals, organizations and businesses.

6. Conclusion and Outlook

Although rarely used, the semantic structure of documents presents a powerful lever to improve conventional sentiment analysis. In this paper, we show how to systematically identify the best parameters for the use of RST trees by performing a grid search. Furthermore, we propose a machine learning approach that significantly improves predictive performance. Our findings reveal that high-level nucleus branches convey the most relevant information.

Future research should investigate methods that consider all levels of the RST tree for sentiment analysis. As such, training an auto-encoder on the full tree structure could be a viable option to handle such a data structure that is not fixed in length. In addition, including all relationship information could be a compelling approach to improving performance further. Finally, it would be very interesting to see how the many different approaches to improving sentiment analysis could be combined into one single best-of-breed approach. To this end a combination of n-grams, negation scope detection, dynamic dictionaries and RST with a suitable algorithm seems very promising, yet challenging.

Acknowledgments

We especially thank Pascal Kuyten for creating the RST trees of our corpus using HILDA.

References

[1] B. Pang and L. J. Lee, *Opinion Mining and Sentiment Analysis*. Hanover, MA: Now Publishers, 2008.

[2] P. Bhatia, Y. Ji, and J. Eisenstein, "Better Document-level Sentiment Analysis from RST Discourse Parsing," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015, pp. 2212–2218.

[3] A. Hogenboom, F. Frasincar, F. de Jong, and U. Kaymak, "Using Rhetorical Structure in Sentiment Analysis," *Communications of the ACM*, vol. 58, no. 7, pp. 69–77, 2015.

[4] J. Hirschberg and C. D. Manning, "Advances in Natural Language Processing," *Science*, vol. 349, no. 6245, pp. 261–266, 2015.

[5] R. Soricut and D. Marcu, "Sentence level discourse parsing using syntactic and lexical information," in *NAACL '03 Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, M. Hearst and M. Ostendorf, Eds., vol. 1, 2003, pp. 149–156.

[6] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, vol. 1631, 2013, p. 1642.

[7] Y. Jo and A. H. Oh, "Aspect and Sentiment Unification Model for Online Review Analysis," in *Fourth ACM International Conference on Web Search and Data Mining*, I. King, W. Nejdl, and H. Li, Eds. ACM, 2011, p. 815.

[8] E. A. Stepanov and G. Riccardi, "Sentiment Polarity Classification with Low-level Discourse-based Features," in *Proceedings of the Second Italian Conference on Computational Linguistics CLiC-it 2015*, 2015, pp. 269–273.

[9] R. S. Trivedi and J. Eisenstein, "Discourse Connectors for Latent Subjectivity in Sentiment Analysis," in *HLT-NAACL*, 2013, pp. 808–813.

[10] Y. Ji and J. Eisenstein, "Representation Learning for Text-level Discourse Parsing," in *Proceeding of the 52nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2014, pp. 13–24.

[11] S. Joty, G. Carenini, and R. T. Ng, "CODRA: A Novel Discriminative Framework for Rhetorical Analysis," *Computational Linguistics*, vol. 41, no. 3, pp. 385–435, 2015.

[12] N. Asher, F. Benamara, and Y. Y. Mathieu, "Categorizing Opinion in Discourse," in *18th European Conference on Artificial Intelligence (ECAI)*. IOS Press, 2008, pp. 835–836.

[13] S. Somasundaran, J. Wiebe, and J. Ruppenhofer, "Discourse Level Opinion Interpretation," in *Proceedings of the 22nd International Conference on Computational Linguistics*, ser. COLING '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 801–808.

[14] M. Taboada, K. Voll, and J. Brooke, "Extracting Sentiment as a Function of Discourse Structure and Topicality," *Simon Fraser Univeristy School of Computing Science Technical Report*, 2008.

[15] A. Lazaridou, I. Titov, and C. Sporleder, "A Bayesian Model for Joint Unsupervised Induction of Sentiment, Aspect and Discourse Representations," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2013, pp. 1630–1639.

[16] W. C. Mann and S. A. Thompson, "Rhetorical Structure Theory: Toward a Functional Theory of Text Organization," *Text & Talk: An Interdisciplinary Journal of Language, Discourse & Communication Studies*, vol. 8, no. 3, 1988.

[17] D. Marcu, *The Theory and Practice of Discourse Parsing and Summarization*. Cambridge, Mass.: MIT Press, 2000.

[18] H. Herneault, H. Prendinger, D. duVerle, and M. Ishizuka, "HILDA: A Discourse Parser Using Support Vector Machine Classification," *Dialogue & Discourse*, vol. 1, no. 3, pp. 1–33, 2010.

[19] P. Kuyten, H. Hernault, H. Prendinger, and M. Ishizuka, "Evaluating HILDA in the CODA Project: A Case Study in Question Generation Using Automatic Discourse Analysis," in *AAAI Fall Symposium: Question Generation*, 2011.

[20] R. Bosman, R. Kräussl, and E. Mirgorodskaya, "The 'Tone Effect' of News on Investor Beliefs: An Experimental Approach," *Available at SSRN*, 2014.

[21] E. Demers, C. Vega *et al.*, *Soft Information in Earnings Announcements: News or Noise?* Federal Reserve Board, 2008.

[22] R. Feldman, S. Govindaraj, J. Livnat, and B. Segal, "The Incremental Information Content of Tone Change in Management Discussion and Analysis," *SSRN Electronic Journal*, 2008. [Online]. Available: http://ssrn.com/abstract=1126962

[23] G. Friesen and P. A. Weller, "Quantifying Cognitive Biases in Analyst Earnings Forecasts," *Journal of Financial Markets*, vol. 9, no. 4, pp. 333–365, 2006.

[24] P. C. Tetlock, M. Saar-Tsechansky, and S. Macskassy, "More Than Words: Quantifying Language to Measure Firms' Fundamentals," *The Journal of Finance*, vol. 63, no. 3, pp. 1437–1467, 2008.

[25] E. Henry, "Are Investors Influenced By How Earnings Press Releases Are Written?" *Journal of Business Communication*, vol. 45, no. 4, pp. 363–407, 2008.

[26] T. I. Loughran and B. McDonald, "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks," *The Journal of Finance*, vol. 66, no. 1, pp. 35–65, 2011.

[27] T. Loughran and B. McDonald, "IPO first-day returns, offer price revisions, volatility, and form S-1 language," *Journal of Financial Economics*, vol. 109, no. 2, pp. 307–326, 2013.

[28] A. K. Nassirtoussi, S. Aghabozorgi, T. Y. Wah, and D. C. L. Ngo, "Text Mining for Market Prediction: A Systematic Review," *Expert Systems with Applications*, vol. 41, no. 16, pp. 7653–7670, 2014.

[29] J. Muntermann and A. Guettler, "Intraday Stock Price Effects of Ad Hoc Disclosures: The German Case," *Journal of International Financial Markets, Institutions and Money*, vol. 17, no. 1, pp. 1–24, 2007.

[30] N. Pröllochs, S. Feuerriegel, and D. Neumann, "Generating Domain-Specific Dictionaries Using Bayesian Learning," in *23rd European Conference on Information Systems (ECIS 2015)*, 2015.

[31] S. Feuerriegel, A. Ratku, and D. Neumann, "Analysis of How Underlying Topics in Financial News Affect Stock Prices Using Latent Dirichlet Allocation," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 2016, pp. 1072–1081.

[32] S. S. Groth and J. Muntermann, "An intraday market risk management approach based on textual analysis," *Decision Support Systems*, vol. 50, no. 4, pp. 680–691, 2011.

[33] N. Pröllochs, S. Feuerriegel, D. Neumann *et al.*, "Detecting Negation Scopes for Financial News Sentiment Using Reinforcement Learning," in *49th Hawaii International Conference on System Sciences (HICSS)*, 2016, pp. 1164–1173.

[34] S. Stoyanchev and P. Piwek, "Constructing the CODA Corpus: A Parallel Corpus of Monologues and Expository Dialogues," in *Seventh International Conference on Language Resources and Evaluation (LREC)*, 2010.

[35] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, Mass.: MIT Press, 1999.

[36] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: A New Benchmark Collection for Text Categorization Research," *Journal of Machine Learning Research*, vol. 5, pp. 361–397, 2004.

[37] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[38] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[39] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[40] T. J. Hastie, R. J. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., ser. Springer Series in Statistics. New York, NY: Springer, 2013.

[41] M. Hagenau, M. Liebmann, and D. Neumann, "Automated News Reading: Stock Price Prediction based on Financial News Using Context-Capturing Features," *Decision Support Systems*, vol. 55, no. 3, pp. 685–697, 2013.

[42] S. Feuerriegel and R. Fehrer, "Improving Decision Analytics with Deep Learning: The Case of Financial Disclosures," in *24th European Conference on Information Systems (ECIS)*, 2016.

[43] N. Pröllochs, S. Feuerriegel, and D. Neumann, "Enhancing Sentiment Analysis of Financial News by Detecting Negation Scopes," in *48th Hawaii International Conference on System Sciences (HICSS)*, 2015, pp. 959–968.