ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG IM BREISGAU

Fachprüfungsausschuss Volkswirtschaftslehre (B. Sc.)

Which news part matters? Using distributed text representations for sentence-level classification of financial news

Bachelorarbeit

Prüfer
Prof. Dr. Dirk Neumann

Verfasser
Maximilan Johannes Hahn
Matrikelnummer: 3748865
Geburtsort: Bad Bergzabern, Deutschland

Bearbeitungszeitraum
18.05.2017 - 18.08.2017

Contents

1 Introduction 4

2 Related Work 5

3 Methodology 7
3.1 Group-Instance Cost Function 8
3.2 Paragraph Vector 11

4 Data Sources 14

5 Results 17

6 Discussion 20

7 Conclusion 22

Bibliography i

List of Figures ii

List of Tables iii

A Appendix iv

Zusammenfassung

Gemäß Artikel 17 der europäischen Marktmissbrauchsordnung sind europäische Unternehmen verpflichtet ihre Aktionäre und die Öffentlichkeit zu informieren, wenn Informationen auftreten, welche möglicherweise Auswirkungen auf vom Unternehmen emittierte Wertpapiere haben. Die Information der Märkte erfolgt dabei durch sogenannte Ad-hoc-Mitteilungen, welche durch spezielle Kanäle verbreitet werden, um eine synchrone Information aller Marktteilnehmer zu gewährleisten. Für den Zweck der Sentiment Analyse haben ad-hoc-Mitteilungen zwei Vorteile. Einerseits enthalten sie klar bewertbare Aussagen, andererseits kann die Wertung eines ganzen Textes durch die Reaktion des Aktienkurses eines Unternehmens auf die Veröffentlichung einer Meldung bestimmt werden. Viele Publikationen versuchen diesen Zusammenhang zu modellieren, um daraus Handelsstrategien abzuleiten [10]. Diese Ansätze erzielen bestenfalls eine Performance knapp über der einer Zufallsstrategie. Die Sentiment Analyse dieser Ansätze beschränkt sich auf die Bewertung der Textebene. Jedoch, ist die Bewertung auf Textebene eine Vereinfachung, denn ad-hoc-Meldungen enthalten oft mehrere Aussagen. Jede dieser Aussagen kann eine andere Wertung haben. Eine Wertung auf Textebene ist damit eine Art Durchschnitt der Wertungen der einzelnen Aussagen in einem Text. Eine genaue Kenntnis der Wertungen einzelner Aussagen in einer ad-hoc-Mitteilung ermöglicht völlig neue Anwendungen. Beispielsweise um die Reaktion von Aktienkursen auf Meldungen besser zu verstehen.
Während die Wertung auf Textebene im Nachhinein durch die Reaktion des Aktienkurses bestimmt werden kann und sich somit ein Trainings-Datensatz erstellen lässt, um einen Klassifizierer auf Text Ebene zu trainieren, ist die Wertung einzelner Aussagen in einem Text auch ex post nicht einfach zu bestimmen. Das in dieser Arbeit präsentierte Model setzt genau hier an. Der Ansatz erlaubt es die Wertung von Sätzen vorherzusagen. Für das Training dieses Satzklassifizierers ist ein Datensatz mit Wertungen auf Textebene ausreichend.
Um einen Satzklassifizierer zu erhalten werden in dieser Arbeit zwei Modelle angewandt. Das Kern Model ist die "Group-Instance cost function" [2]. Diese Kostenfunktion erlaubt es ein logistisches Regressions-

model zu trainieren, welches die Wertung von Sätzen vorhersagen kann. Die Kostenfunktion besteht dabei aus zwei Termen. Der erste Term bewertet die Ähnlichkeit von Sätzen. Werden zwei semantisch ähnlichen Sätzen vom logistischen Regressionsmodel die gleiche Wertung zugeordnet, resultiert das in minimalen Kosten. Ordnet das Model zwei semantisch unterschiedliche Sätze in die selbe Wertungsklasse, entstehen Kosten. Der zweite Term vergleicht die Vorhersagen des logistischen Regressionsmodels für alle Sätze in einem Text mit der tatsächlichen Wertung des Texts. Hierbei entstehen Kosten, wenn die durchschnittliche Vorhersage aller Sätze in einem Text nicht mit der Wertung des Textes übereinstimmt. Durch Minimieren der Kostenfunktion werden die optimalen Parameter für ein logistisches Regressionsmodel ermittelt. Welches die Wertung von unbekannten Sätzen vorhersagen kann.
Die "Group-Instance cost function" ist nicht in der Lage mit Sätzen in gewöhnlicher Textform zu arbeiten. Stattdessen sind Vektorrepräsentationen von Sätzen als Input notwendig. Vektorrepräsentationen von Sätzen sind Vektoren von beliebiger Dimension. Jeder Satz aus einem Datensatz wird mittels eines einzigartigen Vektors dargestellt, wobei die Vektoren so im Vektorraum liegen, dass aus ihrer Lage zueinander eine semantische Ähnlichkeit der Sätze abgeleitet werden kann. Dies geschieht entweder durch die Entfernung von Vektoren zueinander oder durch den Winkel den zwei Vektoren bilden. Sätze sind üblicherweise nicht in Vektorform gegeben, deshalb ist es nötig die Vektoren zu erstellen. Modelle um Vektorrepräsentationen zu erstellen sind ein aktives Forschungsfeld. Aktuell gibt es verschiedene Ansätze dazu. In dieser Arbeit wird der populäre "Document Vector" Ansatz von Le und Mikolov verwendet [6]. Dieses Model kann besonders einfach angewendet werden, da es in der Python Library Gensim [13] implementiert ist.
Um von einem Datensatz, bestehend aus Texten und den zugehörigen Wertungen, auf ein logistisches Regressionsmodel zur Vorhersage der Wertungen von Sätzen zu gelangen, wird in dieser Arbeit folgendermaßen vorgegangen.
Zunächst wird ein Satzvektorenmodel trainiert, um Vektorrepräsentationen von Sätzen zu erhalten. Nun werden die Vektoren und die Textwertungen als Input für die "Group-Instance cost function" verwendet, um ein optimales logistisches Regressionsmodel auf Satz Ebene zu bestimmen. Die Genauigkeit des Models wird dann mit Hilfe eines Test Datensatz bestimmt. Dieser Datensatz besteht aus Sätzen und deren Wertung, welche manuell bestimmt wurde.
Ziel dieser Arbeit ist es, ein Model zur Vorhersage der Wertung von Sätzen aus ad-hoc Meldungen zu erstellen. Folglich wird der oben beschriebene Ansatz an einem Datensatz aus ad-hoc Meldungen überprüft. Die Wertung

eines Textes wird dabei durch den Abnormal-Return des Aktienkurses eines Unternehmens am Tag der Veröffentlichung einer Meldung bestimmt. Da für Ad-hoc Meldungen bisher keine vergleichbaren Methoden auf Satzebene existieren, wird der Ansatz zusätzlich mit einem Datensatz aus Filmbewertungen überprüft.
Die Vorhersagegenauigkeit für beide Datensätze liegt bei etwa 60%. Die Genauigkeit für Sätze aus Filmbewertungen liegt etwas hinter der Genauigkeit anderer Ansätze [2]. Für die Bewertung von Sätzen aus ad-hoc Meldungen gibt es keine Vergleichswerte. Unter Beachtung der Simplizität des gewählten Ansatzes und der offensichtlichen Komplexität die Wertung von ad-hoc Meldungen alleine auf Textebene zu bestimmen, ist eine Vorhersagegenauigkeit von etwa 60% auf Satz-ebene beachtlich.

Abstract

Sentiment analysis is often applied to financial news texts usually to predict stock market movements. Thereby, researchers focus their analysis mostly on text level sentiment. Thus, the structure of sentiment of sentences in financial news texts is unexplored. Knowing the sentiment polarity of each sentence in a text could improve the understanding of how financial news affect stock markets. This work contributes to close this gap by introducing an approach, which allows to predict the sentiment polarity of each sentence in financial news texts. The approach is driven by recent research in the field of natural language processing. The core is a semi supervised learning algorithm which learns a sentence level classifier from text level labels. Experiments show, that the classifier predicts the sentiment of sentences 60% right, so it outperforms two baseline models.

1 Introduction

For European public traded corporations, it is compulsory to provide information to the public, if important news which concern the company occur and this information might have a significant effect on financial instruments issued by the company (Article 17, European Market Abuse Regulation). Thanks to this regulation, there is a vast amount of financial news texts available. These texts come with two benefits, they contain essential information about a company on the one hand and on the other hand the reaction of a company's stock, following an announcement, can be used to easily determine the sentiment polarity of a text. Many researchers try to model the relationship between the sentiment polarity of financial news text and stock market movements [10]. The predictions of such models are at best slightly better than random guessing.

A single financial news text often reports more than on topic, or at least several aspects of one topic. Thus, different sentences in one text are likely to express different sentiment. Sentiment tools applied at text level do usually not consider the structure of the sentiment of sentences in a text. If the structure of sentiment of sentences is meaningful, this is a waist of information.

To better model the reaction of stock markets to financial news, it is crucial to know the sentiment polarity of each sentence in news texts. While the sentiment polarity of a text is determined by the reaction of a company's stock to an announcement made by the company, the sentiment polarity of sentences is unknown.

The approach present in this work can overcome the unknowns. It allows to predict the sentiment polarity of sentences. Thereby, text level labels and vector representations of sentences are sufficient to train a sentence level classifier. While the text level labels are a given fact, vector representations of sentences need to be created. Generating sentence embeddings is a non-trivial task and actuality an active field of research. Hence, there exists no unique and agreed way of obtaining vector representations for sentences [2]. Sentence embeddings in this work are created according to the well-known approach of [6]. This model is implemented in the Gensim Python library [13], which makes it easy accessible.

With vector representations of sentences and text level labels the sentence level classifier is trained using the "Group-Instance Cost Function" (GICF) [2]. To obtain the sentence classifier, the GICF smooth the sentence level classifications based on the similarity of sentences, by also considering the compatibility of the predictions with text labels.

The approach, to predict sentence-level sentiment in financial news text, presented in this work can be easily adopted by other researchers. Helping to better understand the relationship between information in financial news texts and movements of stock prices.

The structure of this work is as follows. In Chapter 2, related literature is discused . Chapter 3 is a description of the overall model, where the "Group-Instance Cost Function" and the paragraph vector model are explain detailed in separated sections. In Chapter 4, the Data Sources and pre-processing are present explained. Chapter 5 shows the results. In section 6 the approach is discussed. Finally, in section 7 conclusions are made.

2 Related Work

The aim of this work is to better understand financial news texts and thus enable to better understand the relationship between the occurrence of financial news and market movements. Therefore, this chapter focus on literature about sentiment analysis in financial news texts and trading strategies based on financial news texts.

The bulk of literature about sentiment analysis, in financial news texts, focuses on trading strategies. First work in that field appeared as early as 1998, [14] uses news, published online, by newspapers like the Wall Street Journal and the Financial Times to predict the closing price of major Asian stock markets. They use a set of keywords, manually created by a financial market expert. The occurrence of the keywords in a specific news text are counted and a model is applied, which predicts whether markets move up, remain equal or move down. The model is trained using the occurrence of keywords in news texts and the belonging market movement of the past 100 trading days. In a three-month period, this approach achieved an average accuracy of 43.6%. In the following years several publications, developing prototypes for predicting financial markets, based on text mining tools, appeared. [10] compared eight such prototypes, where one prototype is the approach of [14]. For the five prototypes, a test result is provided, the accuracy beats a random guess by one fifth up to one half.

Inspired by their findings, Mittermayer and Knolmayer build their own prototype, [9]. As textual input, the prototype uses only company press releases, which are news provided by corporates themselves. As the authors state, press releases come with an advantage over other news texts, published for example by newspapers. This is, regulation requires companies to spread their press releases simultaneously to all market participants. For the usage for text mining, this ensures, that only the initial occurrence of information is used in the model. In contrast, texts published by newspapers often contain no additional information and only comment on press releases. Mittermayer and Knolmayer report, that the average performance of their prototype is roughly 25% above the performance a random trader could achieve.

So fare all presented tools relied on texts from traditional providers of financial related texts e.g. newspapers and companies. But the rise of the web 3.0 leaded to the availability of texts from a different direction. In endless number of blogs people periodically express their own opinion, on diverse topics. [11] were one of the first to use this fact and apply sentiment

analysis tools on financial blogs. They also went beyond the document level classification and applied a finer granularity if one document expresses sentiment to more than one topic. As labels, the authors use a five-point scale between very negative and very positive. They manually labelled 1526 blog entries, each expressing opinion on a company belonging to the Standard & Poor's 500 Index. The classifiers used can predict labels up to 75% accurate.
[15] combined news from both sources, traditional media and social media. Therefore, they took texts from a variety of sources like, blogs, micro blogs and news pages. They perform sentiment analysis in two steps, first they predict text segments containing sentiment signals, then they determine the polarity of the segments. The sentiment predictions are then used to predict abnormal returns and a risk measures for stocks. Overall, the model is not successful in explaining stock market returns, but it shows some interesting correlations, for example between the occurrence of a company in social media and risk of the company's stock.

3 Methodology

The intention of this paper is to develop a model to predict sentence level sentiment in financial news texts. The "Group-Instance Cost Function" GICF [2] thereby is the core of the overall model. The GICF allows to train a classifier, which can predict sentiment at sentence level, by only requiring documents labelled at the text level for training. The inputs of the GICF are not plain texts but vector representations of sentences. To generate vector representations of sentences a second model is necessary. How to best represent sentences as vectors is an active field of research. Vector representations are considered as good, if semantic similar sentences are mapped close to each other in vector space and thus, the semantic similarity of two sentences can be easily determined with distance measures. A brief discussion of existing models can be found in the work of [2]. In this work sentence vectors are created using the paragraph vector model developed by [6]. This is a well-established approach and because of its implementation in the *genism* python library it is fast and easy to adopt [13].
To obtain a sentence level classifier the approach in this paper is the following. First, a model is trained to get vector representations of sentences.

Then the sentence level classifier is trained on the group-instance cost function using sentence vectors and text level labels as input. These two steps are done using a training set with text level labels. Finally, the performance of the classifier is evaluated with a test set which has labels at sentence level. Therefore, vector representations of sentences are inferred from the model trained on the training set. The inferred sentence vectors are then used as input for the classifier, to predict the sentiment labels of the sentences.
In the following sections, the theory behind the two models is described. The group-instance cost function is described with all mathematical details. The mathematical details of the document vector model are above the scope of this work. Therefore, only a general explanation of the underlying idea is given.

3.1 Group-Instance Cost Function

The motivation of the GICF is the circumstance, that there are few texts available with a sentiment classification at sentence level, but there exist much texts with a label at the text level. If there exist no labels at all it is at least easier and less time consuming for humans to provide labels at the text level instead of sentence level. Thus, it would be beneficial, to have a model, which can learn how to spread the document sentiment labels down to sentence sentiment labels. The “Group-Instance Cost Function” can do that [2].
The classifier the GICF optimizes is the logistic regression classifier

$$\sigma(t) = \frac{1}{1+\exp(-t)}. \tag{3.1}$$

The logistic regression distinguishes two classes. Depending on input t, the sigmoid function returns a value between zero and on. For values of $\sigma(t)$ smaller 0.5 the predicted class is negative and the predicted class is positive for values of $\sigma(t)$ bigger 0.5.
In this work, the logistic regression is used to predict the sentiment class of a sentence. Thus, the logistic regression classifier is of the form

$$\sigma(\beta^\top x_i) = \frac{1}{1+exp(-\beta^\top x_i)}. \tag{3.2}$$

The logistic regression depends on the variables $\beta^\top$ and x_i. Where $\beta^\top$ is

the coefficient of the logistic regression and x_i represents a sentence. The classifier cannot use sentence input in text form. Instead each sentence is represented by a vector. In the section 3.2 the method used to create vector representations for this is explained. $\beta^\top$ is the coefficient of the logistic regression. It is a vector with same dimensions as the vector representations of sentences. It ensures, that similar sentences are predicted into the same sentiment class (positive/negative). To find a coefficient, which leads to a good separation of sentences into the different classes is the key task of this work.
The coefficient could easily be estimated with maximum-likelihood estimation, if there would be sentiment labels available for each sentence. This is not the chase for datasets used in this work. Nevertheless, the group-instance cost function allows to find a coefficient $\beta^\top$:
that leads to a good separation of sentences into classes positive and negative. This is how the cost function looks like

$$J(\beta) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{x_i * x_j}{\|x_i\|_2 \left\|x_j\right\|_2}(\sigma(\beta^\top x_i) - \sigma(\beta^\top x_j))^2 + \frac{\lambda}{K}\sum_{k=1}^{K}(\frac{1}{|G_k|}(\sum_{i\in G_k}\sigma(\beta^\top x_i)) - l_k)^2. \tag{3.3}$$

N	the total number of sentences in a training set
x_i, x_j	vector representations of sentences
λ	a tuning parameter
K	the total number of texts in a training set
G_k	the number of sentences in text k
l_k	the label of text k

The function splits into two terms. The first term smoothens the predictions for sentences based on the similarity of sentences and the second term ensures, that the predictions for all sentences in one text are on average close to the label of the text.
In the first term, the similarity of sentences is determined. This is achieved by the cosine similarity of vectors. The cosine similarity calculates the cosine of the angle between two vectors. Thus, ranges between -1 and 1, where -1 indicates that the vectors point in opposite directions, 0 indicates orthogonality and 1 that both vectors point in the same direction. For the vectors A and B the cosine similarity is calculated as

$$similarity = \frac{A * B}{\|A\|_2 \|B\|_2} = \frac{\sum_{i=1}^{N} A_i * B_i}{\sqrt{\sum_{i=1}^{N} A_i^2}\sqrt{\sum_{i=1}^{N} B_i^2}}. \tag{3.4}$$

The first term calculates as the cosine similarities between all sentence embeddings of a dataset multiplied by the squared distance between the results from the logistic regression function for those sentences. Which is normed by the squared number of sentences in the dataset

$$1.term = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{x_i * x_j}{\|x_i\|_2 \left\|x_j\right\|_2} (\sigma(\beta^\top x_i) - \sigma(\beta^\top x_j))^2. \tag{3.5}$$

The second term calculates the values of the logistic regression model for each sentence in a text. The average of these values is a value between zero and one. It is the sentiment prediction of an entire text, based on the sentiment predictions of all sentences in the text. From this text sentiment prediction the actual label of the text is subtracted and the difference is squared. The second term calculates as the sum of this squared differences over all texts in a dataset, normalized by the number of texts in the dataset. The entire second term is multiplied by a parameter λ. This parameter balances the contribution of the second term to the overall costs. It is between zero and one and can be selected freely to fit the GICF to diverse kinds of texts. If λ is set to zero all sentences will be predicted to belong to the same class

$$2.term = \frac{\lambda}{K} \sum_{k=1}^{K} (\frac{1}{|G_k|} (\sum_{i \in G_k} \sigma(\beta^\top x_i)) - l_k)^2. \tag{3.6}$$

To get a reliable β parameter, the model needs to be trained. Training means minimizing the cost function for all observations in a training set. To find the minimum of the cost function, mini-batch gradient descent with momentum is used. Using mini-batches is necessary, because to determine the gradient of the cost function matrix calculations are required. Depending on the size of the input vectors this requires much memory. Thus, it is not possible to calculate the gradient for the whole dataset, using an ordinary computer. Using momentum speeds up the convergence of the cost function.

The optimal β is not necessarily the β observed in the last step of the gradient descent algorithm. Instead, optimum β is selected according to the following rule. In each gradient descent step, the accuracy of logistic regression classifier using the current β is evaluated. After all steps are done, the β leading to the highest accuracy in the validation set is selected as optimum β. The validation is done on text level, where the sentiment prediction for a text is the average of the sentiment predictions of the sentences in the text.

The GICF also has some tuning parameters, allowing to fit the model to different datasets, to get optimal performance. [2] perform a grid search over all parameters and find, that the results are roughly stable for all variations of the parameters. The experiments conducted for this paper confirm that. Different parameter settings change model performance only slightly. A table, showing the exact settings used for the experiments can be found in the appendix.

3.2 Paragraph Vector

The idea of the paragraph vector is to map paragraphs of a text (in our chase those paragraphs will always be sentences) into a vector. The vectors thereby should be able to detect sentences of similar meaning, where similar sentences are represented by nearby vectors. A common and simply method to achieve such vectors is bag-of-words (bow) [5]. Bag-of-words representations can work surprisingly well, but come with some problems. Mainly, they lose the order of words in the sentence and therefore cannot capture all semantic details of a sentence. Also, bow vectors are usually high dimensional. This is because they need a dimension for every word in the dataset. The high dimensional vectors can be problematic if one wants to perform vector operations on them. This is beaus vector operations are memory intense and performing them on high dimensional vectors easily exceeds the available memory of usual computers. The group-instance cost function requires much vector operations, which makes vector representations with less dimensions favourable. The paragraph vector model can overcome these problems. It can represent the semantic meaning of words in a sentence and the meaning of the order in which the words occur. This results in better predicting accuracy of models using this vector representations instead of bag-of-words and other models [6]. Also, the dimension of the vectors can be selected freely, where reliable results are received with 200 dimensional vectors [6].
The paragraph vector was developed by [6] and it is a derivative of the word vector. The word vector model is a neural network, which is described in two research papers [8] [7]. The explaining and understanding of the mathematical details of the neural network is non-trivial. Additionally, Mikolov and colleagues give only brief explanations on the maths in their papers. This makes the paper hard to understand, especially for readers not familiar with the terminology of neural networks. Fortunately, also others find those works difficult to understand, wherefore [4] wrote a paper, where they explain the mathematical details of the original papers. We

do not think that it is helpful if we try to explain again, what has already been explained, also it is not necessary to know the mathematical details of the model to understand our experiments. Thus, we only explain the underlying idea of the models.
The word vector model is a neural network, which learns vector representation of words. It is trained in unsupervised fashion. Successfully trained, the model is capable to detect semantically similar words. Where the similarity of words is measured as the distance of vectors in vector space. As Le and Mikolov state, word vectors can even do more than detecting similar words.

> ...After the training converges, words with similar meaning are mapped to a similar position in the vector space. For example, "powerful" and "strong" are close to each other, whereas "powerful" and "Paris" are more distant. The difference between word vectors also carry meaning. For example, the word vectors can be used to answer analogy questions using simple vector algebra: "King" - "man" + "woman" = "Queen". It is also possible to learn a linear matrix to translate words and phrases between languages.

The general idea of the word vector model is to predict a word given its context. The context is a variable length window of words occurring before and after the word to predict. Figure 3.1 illustrates the underlying neural network.
The figure shows, how the model predicts the word *on* using the context

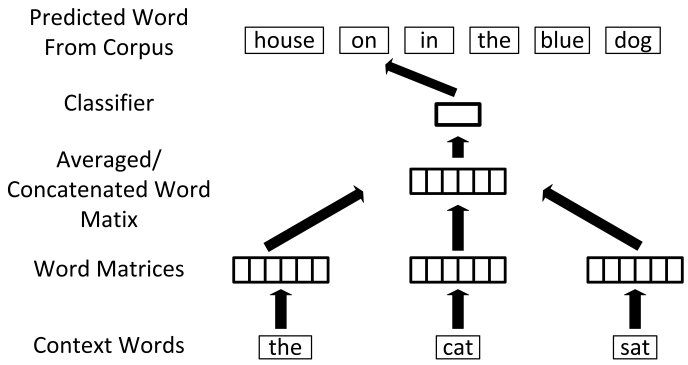

Figure 3.1: Word vector model

the cat sat on. The model input, or the context, are the word matrices of the words *the*,*cat*, *sat* and *on*. These three matrices are concatenated to a single matrix of the same dimension as the word matrices. The concatenated matrix is then used as input in a classification layer. The classification layer predicts which word from all words in a dataset is most likely to appear in the given context.

At the start of the training process, all word vectors are initialized randomly and then trained gradually. In the training phase, the probability to predict the right words given some context is maximized. This is how word vectors are obtained. The model to achieve paragraph vectors is like this.

To get vector representations for paragraphs (sentences in our case) only one a small attachment to the word vector model needs to be done. This is adding in a new vector to the context vectors, which represents the sentence. The resulting model is illustrated in Figure 3.2. The model stays

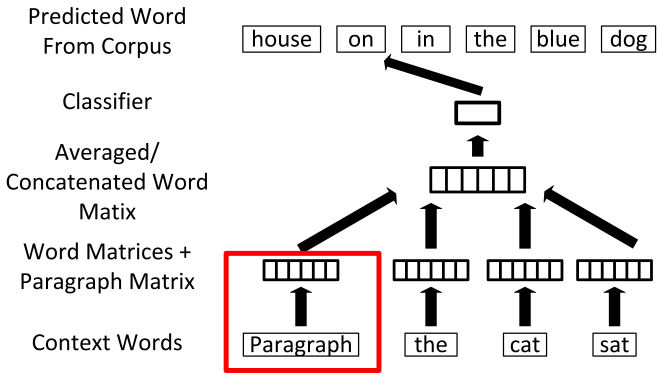

Figure 3.2: Paragraph vector model

the same as in the word model, despite the additional paragraph vector. Now each word is predicted using the vectors of the nearby words and the vector of the paragraph as context. While predicting each word in a sentence, the paragraph vector is the same, while the other context vectors, the words prior to the word to predict, change with every word. Also, the paragraph vector is not shared over the sentences in a text, it is only used to predict one sentence, while the word vectors are shared across the model.

Paragraph vectors obtained that way do represent the semantic of paragraphs well and do perform good as input for a classifier in sentiment analysis tasks. As Le and Mikolov show, the sentiment of movie review texts can be predicted with an accuracy of over 90%, if paragraph vectors trained on whole review texts are used as input.

The paragraph vector model can be adjusted to different tasks with tuning parameters. Because it is an unsupervised learning algorithm, the performance of a model cannot be assessed directly. Instead, the quality of vector representations can only be evaluated through the accuracy of a classification task, where the vector representations are used as inputs.

In this paper, the model settings suggested in [6] are used as a starting point. (The paper does not provide information about all parameter set-

tings, but Mikolov provides access to the original code used[1].) From the model's performance is tried to improve. This is done by varying parameters and assessing the performance through the accuracy of classifications on text-level made by the GICF trained classifier. The parameters used for experiments can be found in the appendix.
Validating how good sentence embeddings do represent semantic similarities of sentences can also be done in a more subjective way as described above. The similarity of sentences can be determined with the cosine similarity of sentence vectors. To find sentences most similar to an example sentence, it is necessary to calculate the cosine similarity between the sample sentence and all other sentences in a dataset. The sentences with the highest cosine similarity measures are considered as most similar to the randomly selected sentence. Figure 3.2.1 shows one example sentence from movie reviews and the three sentences most similar to them. Figure 3.2.2 shows one example sentence from financial news and the three sentences most similar to them.

Cosine Distance	**the only constant is the characters**
0.71	and the dialogue ouch
0.70	the dialogue is sparse
0.70	the basic impotence of the characters only emphasizes the real world difficulties faced by peacemakers

Table 3.1: Example sentences from movie reviews. The four most similar sentences to the first one according to the doc2vec model, with cosine distance as similarity measure

Cosine Distance	**free cash flow in the first nine months of 2009 benefited from a healthy inflow of advance payments**
0.42	deag benefited from tax refunds in the first nine months of 20
0.41	the deterioration is mainly due to lower ebit before one off a higher ramp up in inventory and a lower inflow of advance payments compared to the first nine months of 2008 which benefited from strong inflowing orders at airbus and eurocopter
0.37	cash flow from operations is positive

Table 3.2: Example sentences from financial news. The four most similar sentences to the first one according to the doc2vec model, with cosine distance as similarity measure

1 groups.google.com/forum/#!msg/word2vec-toolkit/Q49FIrNOQRo/J6KG8mUj45sJ

4 Data Sources

We used two datasets for our experiments. A collection of 16459 ad hoc announcements of European public traded corporates is the base for our experiments on financial news. Our movie review dataset consists of 2514 review texts from the Internet Movie Database IMDb.

The main purpose of this work is to develop a model to predict sentence level sentiment in financial news text. The dataset we use for this task contains 16459 financial news texts. In specific, the news texts are ad hoc releases of European public traded corporates, made between the years 2004 and 2011. Ad hoc releases are announcements made by corporates itself. It is compulsory for public traded corporates in Europe, to inform all market participants simultaneously, in chases new information occur, which could affect the valuation of the companies stock. As predictive variable, our dataset contains the abnormal return of a companies stock for the day the announcement was published. The abnormal return measure is gained by subtracting the daily performance of the underling stock index, a company's stock is listed in, from the daily performance of the company's stock. Therefore, the abnormal measures the pure performance of a company's stock, without the noise of general market movements. Using abnormal return as dependent variable is convenient, as it saves the time one would need to classify all news text by hand. But the abnormal return measure comes also with a problem, according to the efficient-market hypothesis [3], stocks allays trade at there fair value, making it impossible for investors to make sustainable profits by using public available information. For our chase of sentiment analysis, this means it should not be possible to successfully predict sentiment with a model trained on data using abnormal return measure as basis for sentiment labels. However, the efficient-market hypothesis is controversial and our data reflects this. The dataset contains numerous observations with high abnormal returns, where it is hard to deny that the market movement was not caused by the ad hoc release itself. But the bulk of the observations have only small abnormal return numbers, making it less likely that the movement was mainly driven by the ad hoc release. To handle these difficulty, we used only the 6000 observations with the highest abnormal returns, 3000 positive and negative abnormal returns. As we conduct a binary classification task, we label the 3000 observation with positive abnormal return measures as positive and the 3000 negative abnormal returns as negative. Out of the remaining observations,

we randomly select sentences and labeled them manually, to report test results. This sentence level testset contains 150 positive and 150 negative labeled sentences.

To test the performance of our overall model, we used a second sort of texts, where already results for sentence level classification exist. This texts are movie reviews. We use IMDb movie reviews for our experiments. The each review text has a also a numerical rating, given by the author of the text, associated. These ratings range from zero to one, split into eleven categories. We again assign binary labels to the observations, observations with a rating higher than 0.5 are labeled positive, observations with a rating lower than 0.5 are labeled negative. This results in 1257 positive and negative labeled movie reviews in the dataset, which we use for training. For testing at sentence level, we use a dataset containing sentences out of movie reviews and sentiment polarity labels for each sentence. This dataset was introduced in [12], and is available online [1].

All texts in our two datasets are formatted like ordinary texts. Which means they use upper chases, punctuation and other things to improve the readability for humans. Like the example text below. To use the texts as input for the *doc2vec model* perform some cleaning steps.

> HKScan Corporation RELEASE 1 October 2008, at 10am
>
> SCAN ACQUIRES INTEREST IN SWEDISH CONVENIENCE FOOD COMPANY
>
> Scan AB, the company responsible for the HKScan Group's Swedish operations, has acquired a 15-percent minority holding in convenience food producer Matfabriken i Skandinavien AB. Based in central Sweden, Matfabriken employs around 100 persons and its net sales in 2008 are projected at EUR 10 million. Its main products include ready-to-eat sandwiches and pasta salads. The company uses its own refrigerated fleet to distribute its products directly to customers, the biggest of which are central organisation ICA, Statoil and SAS. Matfabriken's operative management will retain their positions also after the share acquisition. The transaction includes an option for Scan to acquire the remaining shares in Matfabriken at a later date.
>
> HKScan Corporation
>
> Kai Seikku CEO

The objective of our experiments are sentences. Thus, each text needs to be split into sentences. There are several ways to do that. We use the

1 https://www.cs.cornell.edu/people/pabo/movie-review-data/

PunktSentenceTokenizer available in the *nltk.tokenize python package*. The *PunktSentenceTokenizer* learns in unsupervised fashion how to split a text into sentences. It learns the splitting rules from a large amount of texts of specific topic. We learn the model on all texts in either the movie reviews - or fincancial news dataset and apply the trained model then to each text in the datasets. In the next step, we remove the remaining punctuation from every sentence and replace upper chases with lower chases. As last step, we remove everything, which is not a lower- or upper chase ASCII letter or a number. Numbers are not replaced with a “NUMBER” token. Because, experimends show that a “NUMBER” token influences the paragraph vector model to much, resulting in sentences modeled as similar mainly based on the occurrence of the "NUMBER" token and less on semantic similarity. After the processing the text from above looks like the following:

> ’hkscan corporation release 1 october 2008 at 10am scan acquires interest in swedish convenience food company scan ab the company responsible for the hkscan group s swedish operations has acquired a 15 percent minority holding in convenience food producer matfabriken i skandinavien ab’, ’based in central sweden matfabriken employs around 100 persons and its net sales in 2008 are projected at eur 10 million’, ’its main products include ready to eat sandwiches and pasta salads’, ’the company uses its own refrigerated fleet to distribute its products directly to customers the biggest of which are central organisation ica statoil and sas’, ’matfabriken s operative management will retain their positions also after the share acquisition’, ’the transaction includes an option for scan to acquire the remaining shares in matfabriken at a later date’, ’hkscan corporation kai seikku ceo’,

With this text formatting we can conduct our experiments, which we explain in the next section.

5 Results

Our overall results are better than random guessing. Showing clearly the functionality of our approach. Also, our classifier outperforms the two baseline models we use.
As baseline models we adopt the two models suggested in [2]. This are two easy ways to generate sentence level predictions by training the mod-

els on text-level. One model is a Bag of Words model, where we train a logistic regression model with the BOW representations on text level and apply the classifier then to BOW representations of sentences. In the second approach, we use vector representations of words. The word vectors come as a side product of the *doc2vec* model. They are obtained, like shortly explained in the model chapter above and detailed in these two papers [8] [7]. To get sentence predictions, we trained a logistic regression model on text level, using the average of all word vectors in a text to predict text level sentiment labels. Sentence predictions are then obtained by applying the classifier on the average of all word vectors in a sentence. Table 5.1 below shows the accuracies we obtain for the GICF and the baseline models for our two sentence-level test-sets.

Parameter	**Financial news**	**Movie Reviews**
GICF	0.58	0.61
Word_2_Vec	0.51	0.58
BOW	0.5	0.5

Table 5.1: Sentence level accuracies for the Group instance cost function and two baseline models.

Our classifier predicts the sentiment polarity of sentences most accurate. The word vectors perform slightly worse and the bag of words approach predicts the same class for all sentences.
Figure 5.1 and 5.2 show sentence level performance of the three classifier as ROC plot for financial news and movie reviews.

Our results are satisfactory, but contrary to the results [2] report. They find all three classifier to perform better on sentences from movie reviews, then we do.
Our classifier can also predict text-level sentiment, where the text level sentiment is the average of the sentence sentiment predictions in a text. Table 5.2 shows the accuracies of our classifier and the two baseline models, on text level, for both datasets.

Last, we illustrate how the predictions of our classifier can be applied in analyzing financial news texts. In the example text below sentences are colored according to predictions of our classifier. Sentences in red are predicted negative, sentences in green positive.

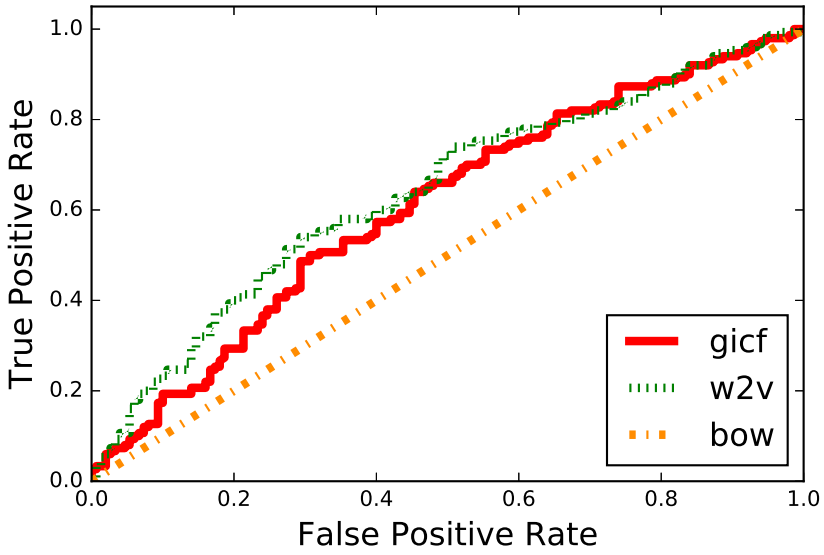

Figure 5.1: ROC curve for sentence level predictions in financial news

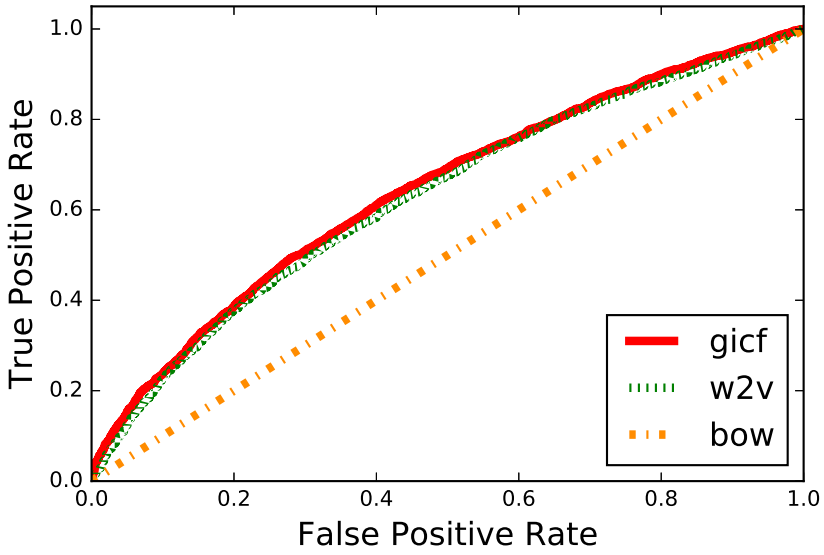

Figure 5.2: ROC curve for sentence level predictions in movie reviews

The share price of princess private equity holding limited princess continues to trade at a significant discount to the net asset value nav of the company, despite seeing a recovery during the past year. The board of directors of princess and partners group, as the investment advisor, have investigated a number of possibilities to address this particular issue with the goal of enhancing shareholder value. The option that is being considered, is a change in the capital structure of the company and providing shareholders with limited but regular liquidity within an open-ended investment company format.

Parameter	Financial news	Movie Reviews
GICF	0.59	0.78
Word_2_Vec	0.64	0.83
BOW	0.82	0.91

Table 5.2: Text level accuracies for the Group instance cost function and two baseline models with average word vectors and Bag of words as vector representations of sentences

For some texts, like the text above, the sentiment predictions of the GICF match the impressions probably most human readers will have. Nevertheless, for most texts the GICF predicts uniformly a positive sentiment for all sentences in a text. Where the overall sentiment of such texts is often also positive, sometimes the GICF is totally wrong. Predicting a positive sentiment for all sentences, where it is obvious for a human reader, that no a single sentence of the text expresses positive sentiment. Like the example texts below show. First text containing only positive sentiment sentences, second text containing only negative sentiment sentences.

> Tria enlarges managing board Munich October 28, 2005 Today the supervisory board has appointed Wolfgang Stuebich 56 as member of the managing board joining board member Richard Hofbauer. Wolfgang Stuebich brings many years of experience in the it and training business to the company. Among others he has worked in management positions or as member of the board for different companies e g Philips data systems digital equipment und broad vision. He will join Tria IT solutions Ag as member of the managing board from November 1, 2005.

> Darmstadt, Germany July 23, 2009 The committee for medicinal products for human use CHMP the scientific committee of the European medicines agency EMEA has today adopted a negative opinion for the use of Erbitux® cetuximab in combination with platinum based chemotherapy for the treatment of patients with epidermal growth factor receptor EGFR expressing advanced or metastatic non-small cell lung cancer NSCLC Merck is evaluating potential appeal options requesting that the CHMP re-examine data demonstrating clinical relevant benefits to patients Steffen Mueller +496151722386 23.07.2009 financial news transmitted by dgap.

6 Discussion

The purpose of this chapter is to discuss the impact of the approach to predict sentence level sentiment polarity in financial news texts introduced in this work on other researchers. Also, this chapter points out possible starting points to improve the accuracy of the model.

Financial news texts often consist of various statements. The different statements in one text are likely to express different sentiment. But another way financial news texts are a collection of sentences where different sentences express different sentiment. Past research focuses on text level sentiment while the sentence level sentiment remains unexplored [10]. Mostly, text level sentiment polarity is used to predict stock market returns. Such models perform only poor.
To know sentiment polarity of all sentences in financial news texts enables researchers to model new unknown relationships between financial news and stock market movements. For example, it could be explored if a company's stock suffers higher losses if bad news appears at the beginning of a news text or of the bad news appear at the end of the text.
Beside trading strategies there are various applications conceivable if the sentiment polarity of sentences is known. For example, an investor could search for all positive or negative news relate to a company. Where positive and negative news could be selected at sentence level rather than text level.
Also, the model presented in this work is capable to predict the sentiment of sentences in financial news texts the accuracy of predictions might be too low to apply the approach for trading or other real world purposes. The following issues can be a starting point to improve the performance.
In this work the sentiment polarity of financial news text in the training set is determined with the abnormal return of a company's stock on a day news where published. As stock markets are usually volatile and noisy this procedure likely results in wrong labelled texts in the training set. Labelling the texts in the training set manually could eliminate this issue.
Purpose of this work is to create an approach which can be easily adopted by others. To ensure this vector representations of sentences are created with a well-known approach which is easy to adopt because of its implementation in a python library. The experiments in this work show that this approach results in slightly worse predictions of sentence level sentiment in movie reviews than in the approach [2] suggest. To adopt the convolu-

tional neural network used in the work of Denil et al. to create sentence embedding's can probably also improve the accuracy of the predictions for financial news texts [1].

7 Conclusion

In summary, research in this paper shows that sentence level sentiment polarity in financial news text can be successfully predicted. Even more, the used approach allows to successfully train a sentence level classifier by only needing text level labels for training.
The model predicts roughly 60% of the sentences in a manually labelled sentence level test set right. To compare the performance of the model with other sentence level approaches experiments with a dataset of movie reviews are conducted. The accuracy of the approach lays slightly behind comparable approaches [2]. The approach presented in this paper is only a first attempt to predict sentence level sentiment in financial news texts. Thanks to its simplicity the approach can be adopted fast and thus enables researchers to further explore sentence level sentiment in financial news texts as well as to improve the accuracy of the approach.

Bibliography

[1] M. Denil, A. Demiraj, and N. DE Freitas. *Extraction of salient sentences from labelled documents*. In: *arXiv preprint arXiv:1412.6815* (2014).

[2] M. Denil et al., eds. *From group to individual labels using deep features*. ACM, 2015.

[3] E. F. Fama. *Efficient capital markets: A review of theory and empirical work*. In: *The journal of Finance*, Vol. 25, No. 2 (1970), pp. 383–417.

[4] Y. Goldberg and O. Levy. *word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method*. In: *arXiv preprint arXiv:1402.3722* (2014).

[5] Z. S. Harris. *Distributional structure*. In: *Word*, Vol. 10, No. 2-3 (1954), pp. 146–162.

[6] Q. Le and T. Mikolov, eds. *Distributed representations of sentences and documents*. 2014.

[7] T. Mikolov et al. *Distributed representations of words and phrases and their compositionality*. In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[8] T. Mikolov et al. *Efficient estimation of word representations in vector space*. In: *arXiv preprint arXiv:1301.3781* (2013).

[9] M.-A. Mittermayer and G. F. Knolmayer, eds. *Newscats: A news categorization and trading system*. Ieee, 2006.

[10] M.-A. Mittermayer and G. Knolmayer. *Text mining systems for market response to news: A survey*. Institut für Wirtschaftsinformatik der Universität Bern, 2006.

[11] N. O'Hare et al., eds. *Topic-dependent sentiment analysis of financial blogs*. ACM, 2009.

[12] B. Pang and L. Lee. *Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales*. In: *Proceedings of ACL*. 2005, pp. 115–124.

[13] R. Řehůřek and P. Sojka. *Software Framework for Topic Modelling with Large Corpora*. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[14] B. Wüthrich et al. *Daily prediction of major stock indices from textual www data*. In: *HKIE Transactions*, Vol. 5, No. 3 (1998), pp. 151–156.

[15] Y. Yu, W. Duan, and Q. Cao. *The impact of social and conventional media on firm equity value: A sentiment analysis approach*. In: *Decision Support Systems*, Vol. 55, No. 4 (2013), pp. 919–926.

List of Figures

3.1 Word vector model 12
3.2 Paragraph vector model 13
5.1 ROC curve for sentence level predictions in financial news . 19
5.2 ROC curve for sentence level predictions in movie reviews . 19

List of Tables

3.1 Example sentences from movie reviews. The four most similar sentences to the first one according to the doc2vec model, with cosine distance as similarity measure 14
3.2 Example sentences from financial news. The four most similar sentences to the first one according to the doc2vec model, with cosine distance as similarity measure 14
5.1 Sentence level accuracies for the Group instance cost function and two baseline models. 18
5.2 Text level accuracies for the Group instance cost function and two baseline models with average word vectors and Bag of words as vector representations of sentences 20
A.1 Group instance cost function parameter settings iv
A.2 Doc2Vec parameter settings iv

A Appendix

Parameter	Financial news
vector size	200
λ	0.05
mini-batch size	5
learning rate	0.05
momentum value	0.7

Table A.1: Group instance cost function parameter settings

Parameter	Setting
dm	0
size	200
window	10
hs	0
negative	5
sample	0.004
iter	20
min_count	10
alpha	0.1

Table A.2: Doc2Vec parameter settings

Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen sind, habe ich als solche kenntlich gemacht. Die eingereichte Arbeit war oder ist weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens. Die elektronische Version der eingereichten Arbeit stimmt in Inhalt und Formatierung mit den auf Papier ausgedruckten Exemplaren überein.

Freiburg im Breisgau, den 01.01.2016