**Introduction**

The prediction of possible stock price changes immediately after news article publications this system consists mainly of four components.

1. The first component gathers news articles and stock prices automatically from internet.
2. The second component prepares the news articles by sending them to some document preprocessing steps and finding relevant features before they are sent to a document representation process.
3. The third component categorizes the news articles into predefined categories,
4. the fourth component applies appropriate trading strategies depending on the category of the news article

This system requires a labeled data set to train the categorization component. This data set is labeled automatically based on the price trends directly after the news article publication. Additional label refining step using clustering is added to improve the labels given by the basic method of labeling by price trends

Experiments showed that the label refining method greatly improves the performance of the system. It was also shown that the timing of when to start the price trends used to label the data sets had a significant impact on the results

**Objectives**

1. study existing systems for automatically analyzing financial news articles with focus on systems that uses the sentiment of news articles in their prediction of future price trends.
2. investigate text mining methods that might be used to create an improved system.
3. design, implement and evaluate a system that uses sentiment analysis on news articles to automatically generate trading signals.

**Hypothesizes**

1. when the stock trading is done from signals generated from sentiment analysis of news articles, then the profit is better compared to what a random trader gives. Or in other words, a news based trader will give positive profits over time.
2. a classifier trained on an automatically created training set performs on the same level as humans at predicting how trends will move after news articles are published.
3. a training set of news articles for the sentiment classifier might be automatically created and labeled by looking at how the price for the related company changes after the article is published.
4. a training set created by looking at price trends after the news article is published is improved by running it by a clustering based algorithm for label refining.
5. the timing of when to start the price trend when it is used for labeling news articles for the training set is important. Starting the price trend a little before the news article is published gives better results since it is certain to capture the early price adjustments right after the news is published

**Stock analysis theory**

Classical technical analysis is based upon three main principles, and they are;

* **market action discounts everything**
  + The statement "market action discounts everything" forms what is probably the most important cornerstone of technical analysis. The technician believes that anything that can possibly affect the price - fundamentally, politically, psychologically, or otherwise - is automatically reflected in the price of the market.
* **prices move in trends**
  + the concept of trends is essential to the technical approach. Technicians say that markets trend up, down, or sideways (flat). Prices move in trends and trends tend to continue until something happens that changes the demand and supply balance. These "changes" can be technical signals such as reversal patterns or breakouts. The goal in this trend-following approach is for the technician to get in on an existing trend as early as possible and ride on it until it shows signs of reversing.
* **history tends to repeat itself**
  + technicians believe that investors collectively repeat the behavior of the investors that preceded them. Because investor behavior repeats itself so often, technicians believe that recognizable and often predictable patterns will emerge. The key to understanding the future therefore lies in studying the past.

**News articles influence on stock markets**

the basic strategy for news-based trading is to buy a stock from companies that has just gotten good news published about them self, or short sell on bad news. Strong and unexpected positive or negative events provide enormous volatility in a stock and gives therefore great chances for quick profits, or losses if they are interpreted wrongly. Determining whether news was "good" (positive) or "bad" (negative) should be determined by the price trend after the news article was published because the market reaction may not match the tone of the news itself. The most common cause for this is when rumors or estimates of the event, like those issued by market and industry analysts, were already circulated before the official news release, and prices have already adjusted them self in anticipation of the official news release.

The number of traded stocks has been shown to be positively or negatively affected by economic news publications (chan, chui, & kwok, 2001). It is also found that both political and economic news articles affect trading activities such as price volatility, number of stocks traded, and trade frequency (chan, chui, & kwok, 2001) this means that there should be possible to create a system that automatically analysis news articles and returns a trade signal (buy, hold, or sell) based the results from its analysis.

**Relevant text mining methods**

Text mining (konchady, 2006) refers to the process of deriving high-quality information from text. High quality in text mining usually refers to some combination of relevance, novelty, and interestingness. Text mining usually involves the process of structuring the input text, deriving patterns within the structured data, and finally evaluation and interpretation of the output. Typical text mining tasks include text categorization, text clustering, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization, and entity relation modeling.

**Preprocessing**

Text preprocessing is the process of making clear each language structure and to eliminate as much as possible the language dependent factors. (wang & wang, 2005) there are many different tasks under preprocessing, but some of the most common ones are tokenization, stop-word removal and word stemming.

**Tokenizing**

Tokenization is the process of splitting a text stream into symbols, words, phrases, or other meaningful elements called tokens. Word tokens are typically sent to preprocessing stages like stop-word removal and stemming, which are described later. They are also used as input for feature extraction processes. There are many ways of tokenizing text streams into tokens. A simple method would be just to split the text on blank spaces, but better methods also take punctuation and other sings into consideration

"hello! This is test number 11. It tests the word\_punct-tokenizer! @ test66"

The tokenized string would then consist of the following tokens.

['hello', '! ', 'this', 'is', 'test', 'number', '11', '. ', 'it', 'tests', 'the', 'word\_punct', '-', 'tokenizer', '! @','test66']

**Stop-word removal**

Stop words are high frequency words of a language that don’t carry any significant information on their own. These words are often removed at the preprocessing stage to reduce the number of features, thus reducing the amount of noise. However, stop words can together with other words contain a significant amount of information. Closed class words like articles, pronouns, prepositions, and conjunctions are usually included in stop-words lists. Some of the more frequently used open class words like auxiliary verbs are also included. It is also possible to create domain dependent stop word lists by filtering out high and low frequency words, or by using some statistical measure like information gain or chi-square to filter out the less informative words. During the removal process all the words that exist in the given stop word list are removed from the source documents.

**Stemming**

In linguistic morphology, stemming is the reduction of a word from its inflected form to its root, stem or base form. It is a common procedure to use in information retrieval, natural language processing and other methods dealing with text analysis to discover the semantic similarity between the different morphological variants of a word. This means that and article that for example uses the word walks and another using the word walking will both have their word reduced to its root which is for both of them walk. By doing this they have both gained a similar feature instead of having two features that the computer would see as totally different even when they for humans clearly have a high semantic similarity. Word stemming has the effect of reducing the dimensionality of features which makes the data less sparse and faster to work with, and that it can be helpful to promote the effectiveness of a text classifier. But some experimental results showed that stemming sometimes might be harmful to the effectiveness of a text classifier(baker & mccallum, 1998). Supporters for some examples of words that are or might be stemmed:

* stemmer, stemming, stemmed -> stem
* cats, catty, catlike -> cat
* fishing, fishes, fished, fisher -> fish

**Features types**

Before text documents can be analyzed by text mining techniques, they must undergo a special processing step known as feature extraction. This process takes the preprocessed text documents and produces a set of features representing each document.

Text features may be surface level lexical features, or semantic or other higher-level features.

* Surface level lexical features are wordbased features that can be observed directly in the documents.
* Semantic features or higher-level features are extracted from the surface level lexical features.

Statistical techniques, such as singular value decomposition (svd), topic modeling, and random projection, are important in solving this kind of problem. Higher-order features can greatly improve the quality of information retrieval, classification, and clustering tasks. These higher-order features are also often used as feature reduction techniques

**Unigrams**

unigrams are n-grams of size one, ore in other words, they consist of one single word. Another name used for unigram features are bag of words feature sets. Feature sets made from unigrams are made of all the selected single words that are left after the documents preprocessing steps. Despite its simplicity, this feature type has proven successful in text classification and word sense disambiguation (mooney, 1996).

**Bigrams**

Bigrams are n-grams of size two. Bigrams are a consecutive sequence of two words, and are very commonly used as the basis for simple statistical analysis of text this might help to improve the classification of the news article sentiment

**Noun phrases**

A noun phrase is a phrase based on nouns, pronouns, or other noun-like words, and it can optionally be accompanied by modifiers such as adjectives. Noun phrases can be used to get more informative features than only single words and less important word are not included as features.

**Proper nouns**

Proper nouns(mark & larry, 2005), also called proper names, are nouns that represent unique entities, such as new york, john smith, or microsoft. They are distinguished from common nouns which describe classes of entities, such as the entities city, planet, person or corporation.

**Name entities**

Name entities are special name entities, such as person names, locations, and organizations, in text documents. It is somewhat like proper nouns, but not as strict. Name entities can also include dates, times, and other numerical information.

**Feature selection metrics (chi)**

Feature selection is crucial to make the classification tasks more efficient and precise because textual data contains a very high-dimensional degree of features. Some common feature selection matrices are information gain, mutual information, odds ratio, term strength, correlation coefficient, and chi-square (tasc & gungor, 2008) (forman, 2003). The chi does not give the best results of the feature selection methods, but it is among the better. It can also work with more than two values.

**Chi-square statistic (chi)**

a chi-square (x^2) statistic is used as a test of independence between each feature and the categories. When the chi-squared value is zero it means that the feature is independent of the category and the larger it gets the more dependent it is on the category. Feature selection can thus be done by only selecting features that has ha chi-square value higher than a given threshold, while the rest of the features can be discarded since they are independent of the categories, which means they have no significance.

Page 25 to see formulas and contingency table

chi-square calculation used has 2 degrees of freedom. The minimum chi value a term can have for it to still be significant is 5.991. If a term has a value less than this it means it is independent, and thus not of any importance when categorizing the documents. This value is found by looking up in a chi-square distribution table where alpha is chosen to be 0.05 and the degrees of freedom that was found earlier is 2.

The chi-square values are normalized , however this normalization breaks down and behaves erratically if any cell in the contingency table is lightly populated (less than five), which is the case for low frequency terms. This is simply solved in this thesis by giving all terms that has a cell with a value less than five a chi value of zero.

**Feature reduction**

Feature reduction is as the name implies the process of reducing the number of features that are used for representing text documents. This is an important process since to many features affect the performance of classifiers and other text mining methods like document clustering and similarity measures

**Singular value decomposition (svd)**

svd is a matrix factorization method which in text mining is usually used in LSA as a well-known and successful dimensionality reduction technique. When svd is used as a feature reduction technique it approximates the initial feature-document matrix by a matrix with a much smaller size. This news smaller matrix does no longer represent the same features as in the original featuredocument matrix but instead it represents latent features

**Vector space model**

Documents are represented as vectors, and each dimension in the document vector corresponds to a unique feature. If a feature does not exist in a document its value is set to zero in the document vector, but if it exist in the document its given a value. This value is calculated by a term weighting method. Feature values are often based on term frequencies and frequency distribution factors. Vector space models make it easy to calculate similarities between documents. It also makes it easy to use each document vector as input for classification an algorithm.

**Term weighting (tf-idf)**

There are three main factors usually accounted for in term different weighting schemas:

* Term (feature) frequency factor,
* Collection frequency factor
* Length normalization factor.

Term frequency factor is the frequency of the term in a given document and collection frequency is the number of other documents in the collection that contain the term. The result from these two factors is important to normalize on the length since text documents vary greatly in length. These three factors are multiplied together to make the resulting term weight.

The term weight is a statistical measure used to evaluate how important a feature/term is to a document. The tf-idf weight increase in proportion to the number of times a feature appears in a document but is offset by the number of number of documents it appears in. The higher tf-idf value a term gets, the more important it is. A high value is reached when the term frequency in the given document is high and when there are few other documents in the collection containing the given term/feature.

**Similarity measure (cosine similarity)**

cosine similarity is used to calculate the similarity between two vectors, document vectors in this case. This is done by measuring the cosine of the angel between the two vectors when the angle between the two vectors are zero it means that they are identical, and the bigger that the angel between them is the more dissimilar they are.

**Classifier learning (svd)**

Support vector machines (svm) (cortes & vapnik, 1995) (vapnik, 2000) are a group of supervised learning methods that performs classification by constructing an n-dimensional hyperplane that optimally separates the data into two categories. Svm models are closely related to neural networks. In fact, a svm model using a sigmoid kernel function is equivalent to a two-layer, perception neural network. Svm has been shown to perform very good on a wide variety of classification problems that require large scale input space, such as handwritten character recognition, face detection, and most importantly in this case, text categorization.

Svm constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest data points in the training set (maximum margin). We want to find the maximum margin hyperplane that divides the two classes

The choice of the kernel function is very important for the efficiency of the support vector machine. Some common kernels are linear, polynomial, radial , sigmoid