AALBORG UNIVERSITY
CASSIOPEIA - HOUSE OF COMPUTER SCIENCE

TENTH SEMESTER REPORT

TAX – Another Approach
Sentimental Analysis

Synopsis:

The aim of the project is to determine whether a document is positive, negative or neutral measured by precision performed in the Danish language. Our approach is a further development of the [Li&Liu,2012] approach, which is a cluster based method where the features are uni-grams and negation. Different calculation methods, feature combination and number of clusters are tested. Our result suggests improvement in comparison with the [Li&Liu,2012] cluster with 10-15% for precision. However, our scores are impacted by the quality of the documents used to label the clusters, which is one of the drawbacks of this approach.

Authors

20030272 Kim Bernhard Andersen
20030276 Jesper Puggaard Hansen
Group d107f13

12. June 2013

Supervisor: Peter Dolog

Table of Content

1 Motivation

In our preliminary work [Andersen & Hansen, 2012] we addressed two main research questions. Firstly we wanted to gain first-hand experience with the different techniques/algorithms within the area of sentimental analysis. Secondly we explored the possibilities of transferring the techniques, which are applied on the English language, to the Danish language.

The main motivation for looking at another approach is that we were not satisfied with the compromise of pros and cons in the different algorithms in the preliminary work. In this report we are going to analyse this issue in order to obtain a better understanding of the requirements for finding a new algorithm/calculation to experiment with. Also, by utilizing the preliminary evaluation data and measurements it is possible for us to make a direct comparison with the new approach, in order to solve the original problem which SKAT gave us.

The algorithms that were implemented and tested in the preliminary work belonged to the rule based class. It also motivates us to look at a different algorithm class to solve the challenge of classifying documents. An advantage of this approach is that we are introduced to other, and new, technologies in the area of concern, which will expand our knowledge of different solutions to the challenge.

1.1 Problem description

Our aim is to create an algorithm able to determine a text's sentimental value. The algorithm should excel in the precision measurement and the target text type is forum posts in Danish. The algorithm should belong to the semi- or unsupervised learning, as small languages do not have as many resources available as a main stream language such as English. The algorithm should handle a noise environment, which a forum is, which means that the algorithm needs to support a neutral classification.

1.1.1 Approach

We have chosen a practical approach in this master thesis, where the focus will be to experiment on, and fine-tune, an existing algorithm, in the hope of improving its performance against common measurement methods and to minimize the pitfalls experienced in our preliminary report. Since we

already have measurements from the preliminary report it would be interesting to perform a comparison with the newly obtained results.

We will carry out an analysis to help us determine the requirements for choosing a new algorithm to experiment with, based on our previous work. We will then implement this algorithm and try to make changes to improve it. We will target the improvement at the algorithm's performance in the classification of documents, either by a new calculation, linguistic features or multiple categories of classifications.

1.1.2 Scope of the report

The analysis performed in this project will be performed at the document level. This report is not about finding the perfect evaluation method. Neither will we look into addressing the language issue in regards to identifying the language. We will assume that the language in scraped posts is Danish, even though we cannot be sure.

We are using libraries within the semantic analysis area, we do not intend to evaluate libraries performance but assume their validity.

The evaluation data is to be the same as in the preliminary report. This enables us to compare the new results with the previous results.

2 Preliminary work

In our preliminary report [Andersen & Hansen, 2012] we were inspired by already existing methods and algorithms performing sentimental analysis applied on the English language. We had two objectives, namely to gain experience within the area of sentimental analysis and to investigate whether the methods could be applied to the Danish language. The algorithms that we had chosen ranged from very simplistic to more complex algorithms and they all belong to the rule based category. The intention with this section is to provide the reader with a summary of our observations and reflections. In conclusion, there will be an analysis of past experience in order to describe possible solutions, since some of these will be addressed further in this master thesis.

2.1 Algorithms

2.1.1 The naïve algorithm

In our preliminary report we presented a simplistic and intuitive approach that rendered a text by identifying positive and negative words based on word lists. This approach caused some difficulties as we were unable to identify any "official" word lists, which stated whether a word was classified as positive or negative. As a consequence we used English words, which were translated into Danish, as well as Danish words that we could categories ourselves. Unfortunately, this approach meant that we were not in possession of complete word lists. Furthermore, as the algorithm determines the sentiment value of a word as either positive or negative this presented us with yet another problem, because a word can both be negative and positive; the sentiment value depends on the context in which the word is used, an issue which this particular algorithm does not address.

2.1.2 The corpus based algorithm

As an attempt to overcome the problem of the context in which a word occurs, we chose to implement a corpus based algorithm. This algorithm still depends on a word list where the probabilities of a word being positive, negative or not classified are calculated. The word list is typically generated by performing a manual classification based upon text from the domain, as generally there are no texts already classified in the specific domain area. The word list cannot have words in them which are not found in the classified texts. Once again, we became aware of the problem of the context in which a word occurs; exemplified by the sentence *"This car is not beautiful"*. This will be interpreted as being positive because of the word *"beautiful"*. The correct interpretation should be negative due to the negation in the sentence caused by the word "not".

2.1.3 The semantic orientation algorithm

The experience gained from the previously implemented algorithms very clearly showed that we had a challenge with the context in which a word occurred as well as with the word lists. In order to overcome this we implemented the semantic orientation algorithm, which uses the assumption that sentiment can be found in adjectives; however these can be both positive and negative. The algorithm requires the ability to identify a word's word class and to manually specify the rules that have our interest. Our own attempt to add a rule was a fiasco, because the construct was never found in any of the evaluation documents. Adding rules requires an understanding of the pattern used for writing by the authors in the forum. The result produced by the algorithms was second best after the corpus based algorithms. The Google version performed best.

2.2 Previous challenges

SKAT has chosen to follow the trend in society and has created a profile on the social network Twitter. The intention of this is to have a fairly simple communication channel for citizens and businesses. However, this requires that the organization is geared to handle the questions asked by the users within a reasonable time frame. To meet this requirement, it is necessary that SKAT allocates the necessary human resources or uses technology that can help identify the most critical or negative questions first. As in any organization economy is a major focus area at SKAT and IT is not a core product, but is viewed as a necessary tool to perform daily tasks. The daily classification task is suited for automation and thereby reduces the cost of the task and this can be achieved with semantics classification algorithms. What SKAT wants is a way to identify the negative opinion, which can be done with semantic analysis (SA). The next challenge is to convince the management that by implementing an automated process, the time spent reading and classifying the tweets can be reduced. Nevertheless, opening a new communication channel still requires a manual process in order to answer questions, either by reference to previous answers or by forwarding the question to the appropriate jurisdiction.

Intuitively we thought it was rather strait forward to shift from one language to another language. However we realized that this is not the case; we ran into our first challenge when we tried to identify Danish word lists, which soon proved to be a nontrivial task. We contacted several agencies to investigate whether such lists existed. Unfortunately the agencies did not possess such lists and they had doubts that any such lists existed. We had no alternative but to try searching the Internet;

unfortunately this did not result in any lists in Danish either. However, we did come across a few word lists in English and we decided to translate these lists and use them in our project. This clearly shows our vulnerability as we could not be sure that our lists were adequate and we had to devote a lot of time to manually scrape the English web pages and translate the words.

The preparation of the training data was conducted in the following way. 205 posts containing 1051 sentences were randomly selected. The random selection means that our web-crawler was crawling the website on a given date collecting the threads created by the users in the order they were created. The evaluation group consisted of three persons who all received an XML document consisting of the selected posts. The evaluators were asked to classify the posts as being positive, negative and neutral/not classified. It is common knowledge that people perceive the content of a text in different ways. Still, it was surprising to observe how differently some of the posts were classified even though the textual content was relatively simple.

3 Problem analysis

3.1 What is sentimental analysis?

SA is a discipline within the area of natural language processing. SA is concerned with establishing the sentimental value of a text, which is often classified into the following two categories: positive or negative. More and more sites allow the user to assign a rating e.g. a movie review from one to five stars. We have created two examples of sentences that contain sentimental bearing words, these are

"I love the movie but nevertheless I hated the main character may he be replaced…."

And

"In theory the movie should have been great even with bad acting"

Interest in sentimental analysis is rapidly increasing because many organizations have realized the value / knowledge that can be inferred from blog posts, news, product reviews and social media. However it is not a trivial task to analyse data created by users as [Pang, Lee & Vaithyanathan] concluded, sentimental classification can be more difficult than text classification since sentimental analysis requires a deeper understanding of the context.

There are many factors that can affect the outcome of the classification. A word can have different meaning depending on the context. Another factor is the role of negation, as described in [Vinodhini & Chandrasekaran, 2012]. Their research has shown that the most common negation words such as: *not*, *neither* or *nor* should be taken into account. Furthermore, attention should be paid to *valence shifters*, *connectives* and *modals*, which are explained below:

Valence shifter: A class of words that changes the semantic value of a sentence. The words can negate, boost/enhance or diminish the meaning of the sentence. Valance shifter is also called modifiers. An example of a valance shifter is

I know what to say

Versus

*I **hardly** know what to say*

In this example the word in **bold** is a valance shifter, because in the first sentence the person knew what to say. This is not the case in the second sentence; where the person was lost for words. Valance shifter is more important to identify on sentence level analysis than on document level evaluation. On sentence level one typically has fewer semantic values than at document level. If one semantic phrase is misinterpreted it will have a large impact on the final verdict because of the low number of semantic phrases to evaluate.

Connectives: Conjunctions [Hatzivassiloglou&McKeown,1997] are the most obvious types (e.g. *and*, *or*, *while*, *because*), but several types of adverb can be seen as connective ("conjuncts" such as *therefore*, *however*, *nevertheless*), as can some verbs (the copulas *be*, *seem*, etc)

Modal: A term used in grammatical and semantic analysis to refer to contrasts in mood signaled by the verb and associated categories. In English, modal contrasts are primarily expressed by a subclass of auxiliary verbs, e.g. *may, will, can.*

Applying this knowledge to the sentences in the example illustrates the word in **bold** belonging to the connectivity group and the underlined word belongs to the modal group.

*"I love the movie but **nevertheless** I hated the main character may he be replaced…."*

And

"In theory the movie should have been great even with bad acting"

3.2 How to perform sentimental analysis?

In order to perform a sentimental analysis two overall methods exist; namely the supervised and unsupervised methods. These methods will be introduced briefly in the following two subsections;

however please note that the intention is to provide a brief overview and not a complete survey of the methods.

3.2.1 Supervised learning

The machine learning approach belongs to the supervised classification, which is also called "supervised learning". The classification requires two sets of documents; a training set and a test set. The training set is used by an automated classifier to enhance the ability to classify a document and the test set is used to validate the outcome of the classifier. Techniques exist, such as Naïve Bayes, maximum entropy and support vector machines, which have achieved great results in the text categorization [Rui Xia, 2011].

3.2.2 Unsupervised learning

In contrast to the supervised learning there is no need for any prior training of the algorithm in order to perform a classification, this is called "unsupervised learning". Several algorithms exist that belongs to this category, e.g. the work of [Turney, 2002], which we have gained experience with through the implementation of the algorithm.

3.2.3 Semi unsupervised learning

According to [Zhu & Goldberg, 2009] semi supervised learning (SSL) is an upcoming learning paradigm that places itself between supervised and unsupervised learning. The purpose of using SSL is to train an algorithm based on labeled and unlabeled data. Using unlabeled data along with a small amount of labeled data has shown an improvement in learning accuracy. This approach can be quite useful within domains where it can be a difficult task to obtain labeled data but where the amount of unlabeled data is plentiful.

3.2.4 Part of Speech Tagger

In order to be able to identify the candidate semantic-bearing words, we have used a Part-of-speech (POS) tagger from [OpenNLP]. The POS tagger works by attempting to annotate words with their corresponding word classes such as noun or adjective. In order to carry out this task, the POS tagger needs a definition that describes the respective word classes. We were able to identify a definition Parole [ParoleDefinition] intended for the Danish language, which is suited to interact with the POS tagger.

Abbreviation	Explanation	Word class
XP	Punctuation	Residual
PP	Personal pronouns	Pronouns
VA	Verbs	Verbs
RG	Adverbs	Adverbs
SP	Prepositions	Prepositions
U	Unique	Unique
PD	Demonstrative pronouns	Pronouns
AN	Ordinary adjectives	Adjectives

Table 1 Explanation of abbreviation POS tagger

In order to provide a better understanding of the described procedure an example is presented. The example is based on the textual content of a post seen in Figure 1 from [amino]. The process of obtaining html pages and the cleaning of noise has been omitted here, but this will be described in

bedømt af 0 Amino'er Dato: 11/17/2005 1:05:19 PM Forfatter: OleGearløs Dato: 11/17/2005 10:15:42 AM Forfatter:destiny Jeg går ikke ud fra at dette er lovligt men nu spørger jeg lige alligevel: Min kæreste har ikke fuldt ud benyttet sit fribeløb for 2005, og så kan jeg jo se en god forretning i at udbetale min løn (fra mit firma) til hende i stedet... Men det er vel ikke lovligt (da hun jo ikke har lavet noget arbejde for firmaet)? Mit råd ville være at hvis du absolut vil snyde, ja så hold din mund om det, og er du ikke kvik nok til at gøre det på egen hånd, så lad være!. Du må tænke på hvor respektløs dit indlæg fremstår over for alle der ikke prøver at snyde i skat. Mvh. Ole Nu var det lige at jeg fik kaffe galt i halsen, for indtil nu har jeg da ikke læst noget om snyderi i denne tråd!? At forsøge at minimere sin skat indenfor gældende lovgivning er da en ret, alle bør benytte sig af!

Figure 1 Post content

section “3.3 Noise in forums”

The result of the first sentence within the example annotated by the POS tagger can be seen as follows:

Forfatter:destiny => XP
Jeg => PP
går => VA
ikke => RG
ud => RG
fra => SP
at => U
dette => PD
er => VA @
lovligt => AN
men => CC
nu => RG

Figure 2 Annotated words

3.3 Noise in forums

The training and the evolution data comes from [amino], which is a public forum where people discuss various issues. This means that our source contains many variants of noise, which will be addressed in this section.

Overall noise

Figure 3 illustrates a screenshot of thread from [amino]. The purpose of the figure is to emphasize the point and it is presented in its unreadable form on purpose. The screenshot has been annotated with a green and red marker to illustrate the noise. This is to symbolise what is considered noise at global level. Most of the page is marked as red, because the information in these parts is either advertisements or components from the website and not from a debater. The green annotation is the only part which the debater has written. This noise never reaches the algorithms, because the noise is removed in a web scraper phase. Therefore the algorithms do not need to handle this kind of noise. The green markers are the definition of a post and a document in our algorithms and it is this text that is being analysed.

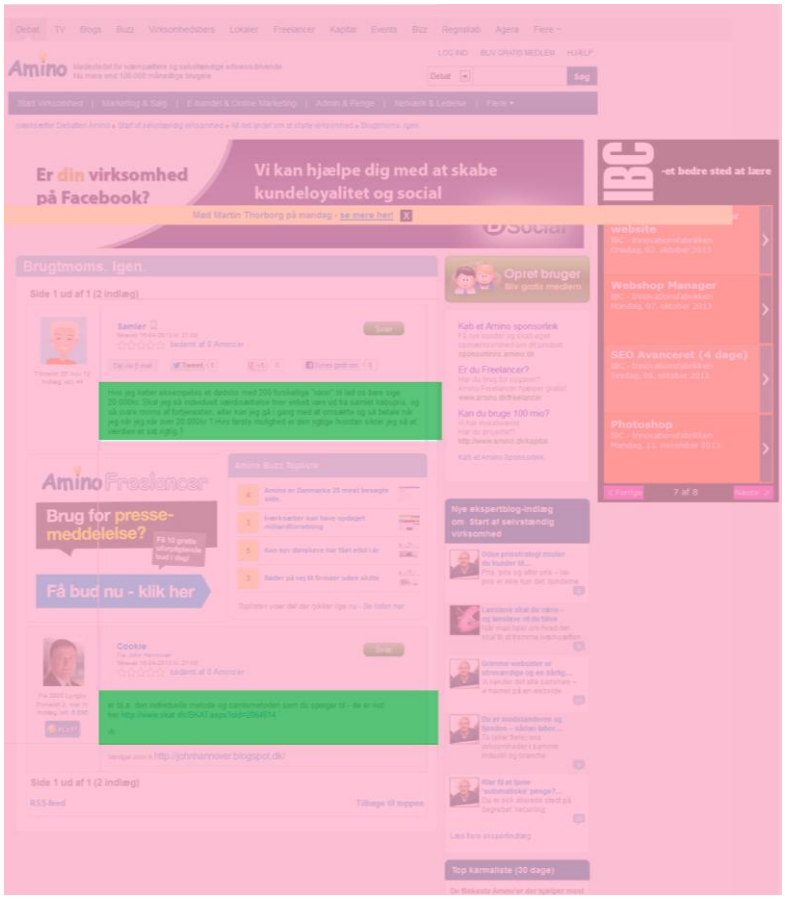


Figure 3 An annotated screenshot from [amino]

Another form of noise is that people do not necessarily write only about the topics dictated by the forum owner. The forum we chose to crawl was about tax subjects, yet many of the collected pages contain other subjects. The resulting problem is that we retrieve and classify posts, which contain data without interest. Furthermore, in the case of a custom dictionary based on a certain topic, this may fail to classify the post correctly. The cause of the incorrect classification is that a word can have different semantic values depending on the specific domain. However, the algorithms in this report are not tailored to the domain, and should not be an issue. This form of noise could be reduced by using information filtering techniques, such as filtering, before sending the post to the algorithms for classification.

Noise in a post

A post can also contain noise. Figure 4 shows a document with noise. The yellow marker shows a quote the debater has included/references and the green marker is the debater's own text. This text is not the debater's opinion, but another person's opinion. This matters if the semantic analysis also needs to find the opinion holder. If this is analysed as one document then the semantic value of the

quoted part will affect the total evaluation of the document. However, in this project there is no distinction between the debater's semantic value and the quoted part. The previous evaluation contained the same noise and as we would like to compare the results we have decided not to remove this kind of noise.

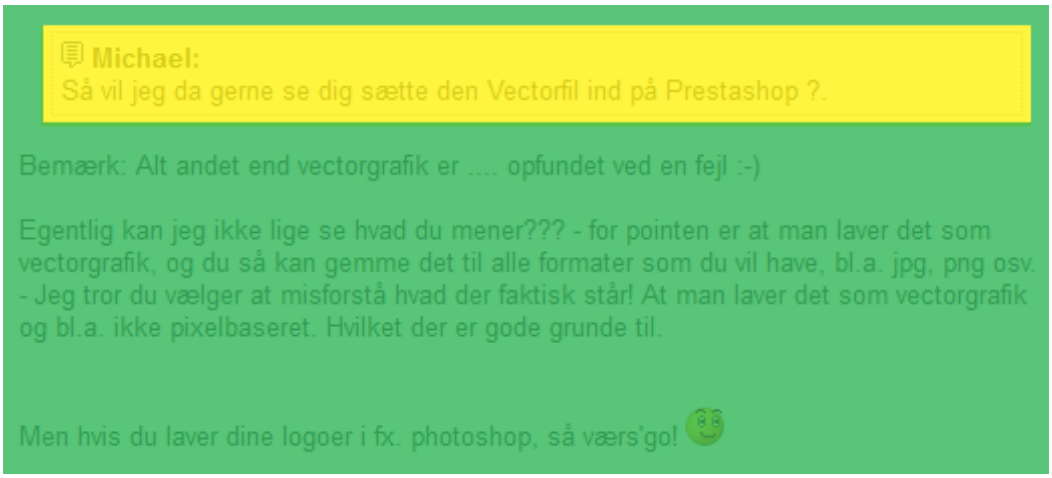


Figure 4 Annotation of noise on document level

Figure 4 also shows another form of noise namely the noise represented as HTML tags and CSS styling. This kind of noise is also removed because this is just a "format" for the debater to communicate the message, not the message itself. The HTML noise is removed in the web scraper phase and in this phase the encoding of the Danish letters is also handled, because there are many ways of presenting the Danish letters in HTML.

Noise on sentence level

Noise is also present on the sentence level. In many cases the [OpenNLP] cannot detect sentences on the documents from [amino]. Some of these errors are due to incomplete definitions of abbreviations in Danish, but some of the errors are due to the writing style of the debater. Many of the sentences are either incomplete, written in spoken language, contain many spelling mistakes and/or is grammatically incorrect. The presented noise is typical for this kind of source; because all types of demographics are writing in the forum. It is not possible for us to change this noise in the material, so the algorithm needs to handle this kind of noise. This also means that algorithms which detect negation, valance shifters and connectives will most likely fail, because it will be difficult to detect the scope to find these linguistic features.

Noise within the algorithm

This kind of noise is created when the algorithm is run in stages and when one of the stages introduces errors of some kind, which the following stage will need to handle. This type of noise is implementation depended, an example of algorithm noise can be found in section 4.1 Clustering method based on TFIDF. In our implementation we generate words with suffix "t", if the word is missing from the synonym list. This basis implementation introduces words which may not exist. The result of this is that the clustering algorithm will have a feature more in the weighted vector. This extra feature should not affect the result because the feature should not have been included in the weighted vector to start with.

3.4 Analysis to identify algorithms

In this section we are looking for new candidate algorithms to experiments with. In order to gain a better decision foundation for selecting new algorithms, we need to understand the issue which we have with classifying our documents as well as the experience of transforming the algorithms into the Danish language from our preliminary report.

Our experience from the preliminary report states that the amount of resources, such as semantics dictionaries in the Danish language, is few and the quality of these are often poor. However, there are many types of dictionaries, such as foreign dictionaries and thesaurus. Perhaps it is possible to find algorithms, which are based on these instead of a dictionary consisting of positive and negative words. The advantage would be that the availability, or presence, is higher as these dictionaries are official but also easier to define. This would be an advantage for "smaller" languages. This also means that algorithms which are not based on dictionaries at all would be an advantage. Therefore our search will be limited to the unsupervised or semi supervised learning.

This has led to the discovery of the article [Li&Liu, 2012], which presents an approach using a synonym dictionary and where the semantic value is determined by the distance from reference words which represents positive and negative. This approach uses a dictionary which we would like to avoid, but in this case it is a synonym dictionary which is much more developed and stable. Furthermore the [Li&Liu, 2012] uses TFIDF in the calculation for classification. This also caught our attention because it is recently published with good results.

The [Martineau&Finin, 2009] also surfaced while researching. This approach suits our needs regarding no dictionaries. The approach only uses one source, it is manually classified documents and the classification is performed on the document level.
The article's algorithms score good results, which makes it a possible candidate. However, the measured results are accuracy and not precision, which is what we are interested in. The reason for choosing precision can be read in "5.1.2 Selection of measurements". It is a gamble to choose this algorithm based on these results; however the accuracy results are so interesting that we think the algorithm is worth pursuing. [Martineau&Finin, 2009] believe they have discovered/ developed a better approach to classifying documents using a variation of TFIDF. This means that we can try to combine these two approaches in order to achieve a higher precision.

We have also looked into the typical document of our domain. A document consists of 5 sentences on average; it is a small piece of text rather than a large article or blog. Our analysis of the noise also suggests that analysis based on correct grammar would properly fail, because of the construction of the sentences. Also, our experience tells us that the sentence detector is having a difficult time with this material. These are variables we need to handle in one way or another.
The small number of sentences suggests that adding a valance shifter feature would help in achieving a high precision of classifying documents. The reason for this assumption is that the number of semantic words that are misinterpreted will have a large impact on the final verdict. As our goal is to improve the precision of the algorithm, we believe we need some kind of valance shifter feature in the algorithm. Valance shifter research resulted in articles about the subject but not in a complete approach. The closest was [Asmi&Ishaya,2012] and this approach is also the only approach which do not use a dictionary to identify negation words.

Another fact about our documents used for classification is that many of the documents do not contain any semantic value at all. The evaluation dataset contains 39% neutral documents. It is a requirement that this type of document can be identified, because otherwise the precision for the positive and negative class will suffer. The articles [Li&Liu, 2012] and [Martineau&Finin, 2009] only describes the positive and negative class. [Li&Liu, 2012] can only identify these documents if the documents do not contain any adjectives or adverbs. However, not all adjectives or adverbs contain semantic value. The [Martineau&Finin, 2009] do not write about neutral documents, but a

threshold limit could be added. Our idea is to change the [Li&Liu, 2012] algorithm to be able to generate 3 clusters instead of 2 in our attempt to improve precision.

The three mentioned articles will be the basis for our approach to create an algorithm with the highest possible precision for this domain and with the lowest demands for human and economic resources. We have chosen to use [Li&Liu, 2012] as our base since we believe that it is easier to incorporate the negation into the algorithm and because we experiment with different numbers of clusters and different weight calculation methods.
The algorithm will be described in details in section "3.5 Algorithms for research".

3.5 Algorithms for research

This section describes the original algorithm from the paper we were inspired by. In later sections the different parts will be used to create our implementation of the algorithm.

3.5.1 Clustering method on sentiment analysis algorithm

This algorithm was introduced in the paper [Li&Liu, 2012]. The idea behind this algorithm is to use a clustering method to classify the documents. The algorithm is divided into multiple stages, which also makes it very flexible and easy to change/improve upon. The stages can be divided into

1. Pre-processing of the documents
2. Converting the documents to vectors
3. Apply weight to the vectors
4. Execute multiple runs of the clustering algorithm
5. Find the result by using a voting mechanism upon the multiple cluster runs

Pre-processing of the documents

The first stage is based on preparing the documents by finding the feature which the result should be based on. In the original algorithm the features consisted of words that belongs to either the word class adjectives or adverbs. Furthermore, the words are stemmed to reduce the number of words, thereby reducing the number of features in order to reduce the calculation time for the clustering algorithm. However, it is possible to enhance the algorithm to include further features, such as negation or linguistic features [Gamon, 2004]. This is an example of the flexible nature of this approach. The original implementation only uses surface features. Removal of noise could be

carried out at this stage or it could be carried out in advance. The paper [Li&Liu, 2012] does not mention anything about removing noise.

Converting the documents to vectors

In stage two a vector (list) is created for each document in the corpus and the preselected documents. The preselected documents are called *seed documents*, which are documents with a known classification. The seed documents are used in a later stage, where they are used for determining the sentiment value on the cluster. The length of the vector is determined by the number of words found in the new corpus called *execution corpus*, which combines documents of the corpus and the seed documents. The end result of this stage is a matrix, which has the number of rows that equals the number of documents in the *execution corpus* and the number of columns equals the number of features/words found.

Apply weight to the vectors

In stage three each feature, such as an n-gram, is given a weight. The original paper presents two methods of calculating the weights of a vector. The calculation consist of the following elements

1. The **TFIDF value** for a word *w* in the document *d* of the corpus *c*
2. A **distance weight**.
    a. The weight is a sub calculation, which is based on the minimum distance [1](D) between the word *w* to the reference words for the positive and negative sentiment. The reference values used by [Li&Liu, 2012] are good and bad. The distance is found by using [Wordnet] as a source for the distances between words. An example is depictured in Figure 5. Here the shortest path is the one with green arrows, which has a distance of 2 to good. The other distance to bad is shown with red arrows and has a distance of 3. It is unclear from [Li&Liu, 2012] if the distance is calculated on the synset level or on the word level; this has not been clearly defined by the author. A synset, or synonym ring, is a group of elements with the same semantical meaning, in the area of information retrieval.

[1] The minimum distance has been confirmed by the author of the paper as it was unclear in the paper

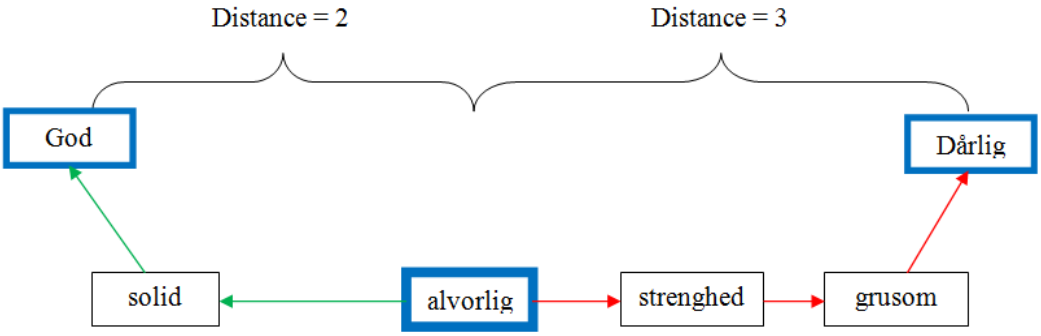


Figure 5 Example of distance graph for the word "alvorlig" (serious)

b. The calculation is

$$distance\ weight = \begin{cases} 1.2 - (\mathrm{D} - 1) * 0.02 & if\ D \leq 8 \\ 1 - (D - 1) * 0.1 & if\ 8 < D \leq 11 \end{cases} \quad (1)$$

3. The last component is **existing weight,** which is either the frequency or the presence of the word *w* in the document. This is the difference between the two calculations.

The complete formula is

$$\boldsymbol{vectorWeight} = TFIDF(w,d) * distanceWeight(w) * \mathrm{existingWeight(w,d)} \quad (2)$$

The end result of this phase is a matrix consisting of the documents on one axis and the different features on the other axis. The cells contain the vector weight and are illustrated in Table 2.

	Good - Feature1	**Bad - Feature2**	**Beautiful - Feature3**
Document 1	Weight11	Weight21	Weight31
Document 2	Weight12	Weight22	Weight32
Document 3	Weight13	Weight23	Weight33

Table 2 Illustrate the matrix created for the k-mans clustering

Execute multiple runs of the clustering algorithm

In stage four the clustering algorithm is executed with the generated weighted vectors. The clustering algorithm k-means is used in [Li&Liu, 2012] and the clustering distance measurement used is the cosine based. The centroids are randomly selected. This creates two clusters. The clusters need to be identified as positive or negative. This is done by using the seed documents. The following test in Pseudo code 1 is used to determine the cluster classification

1. **If**

$$\text{count}\big((\text{Pos seed}) \in \text{Cluster}_0\big) > count\big((\text{Pos seed}) \in \text{Cluster}_1\big)$$

and

$$\text{count}\big((\text{Neg seed}) \in \text{Cluster}_0\big) < count\big((\text{Neg seed}) \in \text{Cluster}_1\big)$$

Then

$\text{Cluster}_0 = \text{Positive}$
$\text{Cluster}_1 = \text{Negative}$

2. **Else If**

$$\text{count}\big((\text{Pos seed}) \in \text{Cluster}_0\big) < count\big((\text{Pos seed}) \in \text{Cluster}_1\big)$$

and

$$\text{count}\big((\text{Neg seed}) \in \text{Cluster}_0\big) > count\big((\text{Neg seed}) \in \text{Cluster}_1\big)$$

Then

$\text{Cluster}_0 = \text{Negative}$
$\text{Cluster}_1 = \text{Positive}$

Pseudo code 1 assigning label to cluster

Since the centroids are randomly selected the result of the clustering is unstable, which impacts the classification of the documents. To stabilize the result the author of the [Li&Liu, 2012] has chosen to run the clustering several times and use the result for the voting mechanism in the last step to generate the final classification of a document. The suggested number of cluster runs is 20 times.

Find the result by using voting on the multiple cluster runs

The last stage is to determine the final classification for a document based on the executed cluster runs. The test used to determine the final classification is

$$\boldsymbol{Positive} = count(d \in positive\ ClusterRuns) > count(d \in negative\ ClusterRuns)$$

$$\boldsymbol{Negative} = count(d \in positive\ ClusterRuns) < count(d \in negative\ ClusterRuns)$$

Pseudo code 2 finding the end result

3.5.1.1 Selection of seed documents

To use the algorithm one needs to identify seed documents with a high quality. There are two suggested ways of finding seed documents by [Li&Liu, 2012]. The first method is to create two documents. One document should only contain very positive words and the other should only contain very negative words. However, the manually generated documents are often misclassified, which impacts the performance according to [Li&Liu, 2012].This is the reason the second approach has been chosen, since in this approach the documents are very rarely misclassified according to [Li&Liu, 2012] experiments.

The second approach is to manually classify a number of real world documents. Then select an equal amount of positive and negative documents and execute the first four stages of the algorithm without adding the seed documents. Then the cluster runs are set to 100 iterations, this is done in order to find the most stable seed documents. The cluster classification is determined by calculating the accuracy of the two clusters.

	$Cluster_0$	$Cluster_1$
Positive	a	b
Negative	c	d

Table 3 Confusion table for labelling clusters

The formula is based on accuracy presented in equation number (3) and this is depictured in Table 3

$$accuracy \begin{cases} \dfrac{a+d}{a+b+c+d} & if\ (a+d) \geq (b+c) \\ \dfrac{b+c}{a+b+c+d} & if\ (a+d) < (b+c) \end{cases} \quad (3)$$

This should be understood as

$$(a+d) \geq (b+c) = \begin{cases} Cluster_1\ is\ Positive \\ Cluster_0\ is\ Negative \end{cases} \quad (4)$$
$$(a+d) < (b+c) = \begin{cases} Cluster_1\ is\ Negative \\ Cluster_0\ is\ Positive \end{cases}$$

This means that the highest accuracy dictates the labelling of the clusters. Now the clusters are labelled. The seed documents are documents which are never classified wrongly or at least only a very few times. They can be found by comparing the manual classification of the document with the cluster classification of the document. This means that if a positive manually classified document is found in the positive clusters all 100 times it is a very positive document and it will rarely be classified as a negative document. The reverse is true for the negative documents.

3.5.2 Delta TFIDF

Delta TFIDF is another approach used to calculate the sentimental value of an n-gram. Typically the weight indicates how rare or how common the n-gram is to the document. However, the idea with delta TFIDF is to weigh how biased an n-gram is to a document. This calculation boosts the terms which are unevenly distributed between the positive and negative documents. This means that an n-gram that appears equally in the positive and negative documents will get a weight of 0. The score will be higher the more prominent the n-gram is in one classification than in the other, however in the opposite direction. If the n-gram is very prominent in the negative classification, then the n-gram will have a high positive score. If the n-gram is difficult to find in the negative documents then the disadvantage to enhance the value is very small and a good tell-tale sign that the document is positive and boosting this n-gram will only affect negative documents minimally.

The calculation of Delta TFIDF is

- $C_{t,d}$ is the number of times where term t occurs in document d
- P_t is the number of positive documents with the term t
- N_t is the number of negative documents with the term t
- $|P|$ is the number of positive documents in the training set
- $|N|$ is the number of negative documents in the training set

$$V_{t,d} = \left(C_{t,d} * log_2\left(\frac{|P|}{P_t}\right)\right) - \left(C_{t,d} * log_2\left(\frac{|N|}{N_t}\right)\right) \quad (5)$$

$$V_{t,d} = C_{t,d} * log_2\left(\frac{|P|}{P_t}\frac{N_t}{|N|}\right)$$

When the training sets are balanced the formula can be reduced further to

$$V_{t,d} = C_{t,d} * log_2\left(\frac{N_t}{P_t}\right) \quad (6)$$

3.6 Valance Shifters Algorithms

In this section we will describe some techniques to identify and evaluate valance shifter. We have chosen two techniques as we found the polarity calculation interesting and matching our overall criteria's and "Bag of words" (BOW) is a very simple technique, which is the reason for choosing this.

3.6.1 Bag of words Negation

BOW is a simple supervised technique used to identify words. The words can be positive, negative or any other type. In this case we are looking at BOW for negation words. This means that in this case the lexicon contains negation words. The lexicon is used as a source to identify negation sentences. When one of the words is found then all the rest of the words in the sentence will be in the scope of the negation. This is done by adding NOT in front of the words, like so

*I do **not** like the new BMW model*

Transformed into

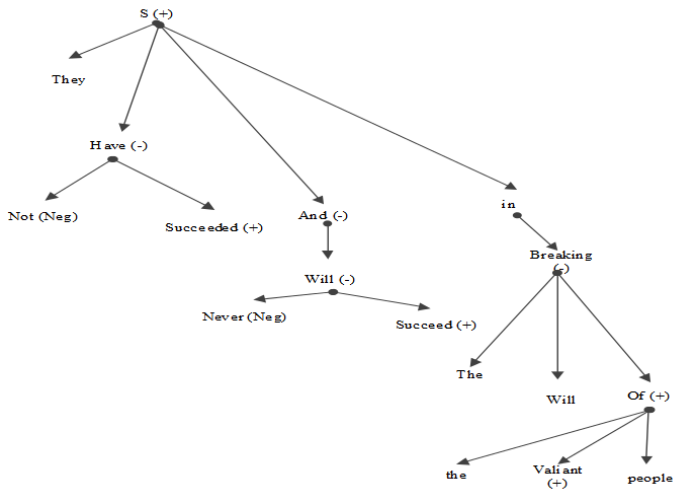
I do not like [NOT] the [NOT] new [NOT] BMW [NOT] model

This means that the words after a [NOT] token should switch semantic value. A typical formula for this is simply to switch the sign of the semantic value of the word following the [NOT] token. This technique is only dealing with negation and not with the diminisher/booster from the concept valance shifters.

3.6.2 Polarity Calculation

The polarity calculation method is part of a bigger framework determining the semantic value of a text. In this case we are only interested in the negation part of the framework in which we have found our inspiration. The framework uses Stanford syntax parser [Stanford] and a [Penn Treebank] POS tagger to extract information from the text. [Stanford] has the ability to identify sentences that contain negation, which the author of [Asmi&Ishaya,2012] use in the polarity calculation method. The scope of the negation is found by using [Penn Treebank] by identifying the noun and verb phrases in a sentence. This means the scope is based on phrase level because the technique is based on building a tree structure of the sentence. This example is taken from [Asmi&Ishaya,2012] and the sentence is

"They have not succeeded and will never succeed in breaking the will of the valiant people"




```
(Sentence
  (Pronoun They)
  (Verb Phrase
    (Verb Phrase (have not)
      (Verb Phrase (Verb succeeded))
    (and)
    (Verb Phrase (will))
      (Adverbial Phrase (Adverb never))
      (Verb Phrase (succeed))
    (Prepositional Phrase (in)
      (Sentence
        (Verb Phrase (breaking)
          (Noun Phrase
            (Noun Phrase (the will))
            (Prepositional Phrase (of)
              (Noun phrase (this valiant people))
```

Figure 6 Example of scope of valance shifters

On the left side in Figure 6 a dependency tree is generated over the sentence and on the right side an abstract syntax of the tree is presented. The tree is generated by the POS tagger and one can see that

the POS tagger has found several phrases. With this example it is possible to see that the negation word "not" in the verb phrase "have not succeeded" is not impacting the rest of the sentence.

Sentence level is calculated based on a calculation and a rule set. The rule set can be seen in Table 4

First Word/Phrase/Clause	Second Word/Phrase/Clause	Negation present	Sentiment Result of the combination
Positive	Positive	True	Negative
Positive	Positive	False	Positive
Positive	Negative	True	Positive
Positive	Negative	False	Negative
Negative	Positive	True	Positive
Negative	Positive	False	Negative
Negative	Negative	True	Negative
Negative	Negative	False	Positive

Table 4 Article rule set for negation

The formula for calculating the semantic value is

$$Resulting\ Intensity = \quad (7)$$
$$First\ Word/Phrase/Clause + [(1 - Second\ Word/\text{Phrase}/Clause) * Second\ Word/Phrase/Clause]$$

The calculation is done from the leaf level up to the root. This means that first the word's polarity is calculated, then the phrase and clause level and at last the sentence level. The semantic value for a word is retrieved from [SentiWordNet] and if negation is present the sign is changed on the value. Figure 7 contains the pseudo code of the calculation.

```
1.  Function CalculatePolarity Returns Polarity{
2.      Double polarity = 0
3.      For Each nounPhraseOfSentence {
4.          For Each word Of type Noun and Adjective {
5.              var sentiValue = getSentiWordNet(word)
6.              If (sentence is Marked NEGATION by Syntax Parser) {
7.                  sentiValue = -sentiValue
8.              }
9.              polarity += [( 1 – Noun/Adjective) * Noun/Adjective]
10.         }
11.     }
12.
13.     For Each verbPhraseOfSentence {
14.         For Each word Of type Verb and Adverb {
15.             var sentiValue = getSentiWordNet(word)
16.             If (sentence is Marked NEGATION by Syntax Parser) {
```

```
17.
18.              sentiValue = -sentiValue
19.             }
20.             polarity += [( 1–Verb/Adverb) * Verb/Adverb]
21.         }
22.     }
23.
24.     Return polarity
25. }
```

Figure 7 Polarity Calculation

3.7 Contributions

One of our contributions is generally to make the algorithms work on Danish. More specific contributions are to enhance the "clustering method on the sentiment analysis" algorithm. We will try to enhance the algorithm in three main areas. One of the three changes is another way to calculate the metrics for the clusters, by combining the calculation with Delta TFIDF and completely override it with Delta TFIDF.

The next enhancement is to investigate the ability of adding multiple clusters instead of two. The reason for this enhancement is to improve precision of the not classified/neutral category, because with the current implementation not classified/neutral can only be detected if no semantic words are found.

The last enhancement is to improve the algorithms' understanding of valance shifters. This is done in order to improve precision for negative and positive documents, but also to investigate how difficult it will be to find semi- or unsupervised methods to handle valance shifters. The evaluation documents are quite small, which means that misinterpretation will impact the precision.

A side contribution is to rerun our experiments for the corpus based algorithm with a different dataset for training than previously. This is to determine if the evaluation data can be used as training data without an issue.

4 Problem solution

This section will describe the changes made to the algorithm in the 3.5 Algorithms for research section

4.1 Clustering method based on TFIDF

The first implementation is very close to the original algorithm and will be named *ClusteringDanish*. Our modification consists of these changes

- **The source of the distances**

  The algorithm needs to work on the Danish language but [Wordnet] does not contain the Danish words. In Denmark a similar project to [Wordnet] exists, this is called [Dannet]. Like [Wordnet], [Dannet] also works with synset. Therefore it was a logical choice to use [Dannet] as the source. After investigating the content of [Dannet] it was soon discovered that the Danish version contained very few synonyms in each synset and many synsets did not have any synonyms at all. It was evaluated that the content was too small, because our preferred reference word did not have any synonyms. This issue affect the algorithm since the distances could not be calculated.

  Then we searched the Internet for a different source and found the Open Office synonym dictionary, which contains around 12.000 words and around 14.400 synonym definitions. The words have been persisted in a database and the distance to the reference words “godt” (good) and “dårlig” (bad) has been calculated in advance and stemmed. We discovered that the implementation of [Lucenestemming] is not being carried out on words that end with a “t” such as “gammelt” (old), “præcist” (precise) and so on. This is not necessarily an issue, but the problem is that the synonym dictionary didn’t contain these words, which results in losing input to the clustering algorithm. In order to improve on the input we chose to try to improve the dictionary in our table. The implementation is crude in the sense that it looks for words not ending with “t” and which does not contain any spaces. The new word definition is a duplication of the original word, except the value for the word column and stemming column is the original word appended with a “t”. An example of this is the word “gammel” (old), it will be found as it does not contain “t” nor contains a space. A new entry will be created for the new word “gammelt” (old) with the same value. However, this

approach does not guarantee that the outcome is a dictionary word or that the word has the same semantic meaning as the original word.

In the situation where the word is not a dictionary word, it will most likely not be found in the source document. If it is, the person will most likely be using it with the same logic/rule for appending the "t". There is a very high probability that we have captured the meaning, which otherwise would have been lost.

The second situation is more of a problem. An example of this is that the word "to" (two) would become "tot" (wad/turf), which has a completely different meaning. Some of these situations are mitigated because if the words already exist in the table no changes are made. However, if this is not the case then there is an issue. The problem is that the distance value saved on the record is from the original word. This situation will create noise for the clustering part of the algorithm. This situation could have been reduced by checking the words in the same synset or if the distance had been based on synset instead.

The max distance is 11 for any of the reference words.

In our implementation the distance is calculated on the word level as the Open Office synonym dictionary does not contain synsets. Also the Open Office synonym dictionary doesn't contain word classes on the synonyms. This means that we jump word classes when calculating the distance.

- **Multiple classifications categories**

  Our old results have three classification categories (positive, negative, neutral/not classified) and we would like to compare the new result with these. Two solutions exist to retrieve the third class. The first solution is to classify all the documents with neutral/not classified if the documents do not contain any adjectives or adverbs, then we have no words to create the weighted vectors from. This will only result in a 0 based weight on all of the features.

  The second solution is to generate not just two clusters, but to generate multiple clusters instead. We have achieved this by adding seed documents with the new classification category and changing the labelling test in stage four. The test has been changed to find the cluster name, which contains the most seed documents of that class. The found cluster name for each class is checked for uniqueness. This has been illustrated with examples in Table 5 and Table 6.

	$Cluster_0$	$Cluster_1$	$Cluster_2$
Positive	X		
Neutral/not classified		X	
Negative			X

Table 5 Example of a situation where the clusters can be labelled in multiple cluster generation

	$Cluster_0$	$Cluster_1$	$Cluster_2$
Positive	X		
Neutral/not classified		X	
Negative	X		

Table 6 Example of a situation where the clusters cannot be labelled because we cannot figure out if $cluster_0$ is positive or negative

4.1.1 Variations

We have created some variations of the clustering algorithm. The variations change how the weight of the vector is determined. The variation called *Delta TFIDF* uses the calculation described in section 3.5.2 Delta TFIDF. The last variation created is also based on the Delta TFIDF calculation but this also takes the distance aspect into the calculation, just like the original algorithm. This variant is called *Distance Delta TFIDF*. This variation is a hybrid of the two calculations. The logic behind trying this variation is to observe if the distance scale interferes with the Delta TFIDF scaling of goodness or badness, or if they can co-exist and enhance each other. The formula for the different variations can be viewed in Table 7.

Algorithm name	Formula	
ClusteringDanish	$TFIDF(w,d) * distanceWeight(w) *$ existingWeight(w, d)	(7)
Delta TFIDF	$DeltaTFIDF(w,d) *$ existingWeight(w, d)	(8)
Distance Delta TFIDF	$DeltaTFIDF(w,d) * distanceWeight(w) *$ existingWeight(w, d)	(9)

Table 7 Weight formula for the different variations

4.2 Valance shifter Feature

We have chosen to add the negation feature as an input to the algorithms. Our design is based on the inspiration from the negation algorithm described in "3.6 Valance Shifters Algorithms". Our approach is to target valance shifter both for negation, diminisher and booster, but not for modal expressions. Basically the design can be broken down to three components, which we need. The

components are used to detect the situation, the scope of the impact and how to impact the semantic value.

We have chosen two different approaches for negation and diminisher/booster. These approaches will be described in the next sections.

4.2.1 Negation

One of our overall goals is to create an algorithm belonging to either semi- or unsupervised learning. This is also the goal for the negation feature; however it has been difficult to find. [Asmi&Ishaya,2012] seems to be unsupervised for detection of the negation. According to the article the Stanford syntax parser can mark the words as being a negation word or not. We have researched whether this tool is supported in Danish. The tool can be used on the Danish language, if one creates a Danish grammar parser. To our knowledge no Danish grammar exists for the [Stanford] tool.
This has forced us to use a lexicon based approach to detect negation, which is more like the BOW. We also have an issue with the scoping because [Asmi&Ishaya,2012] uses a Penn Treebank POS tagger to determine the scope of the negation. The Penn Treebank POS tagger is used to find phrases such as noun phrases. Our Parole POS tagger does not support phrase level identification. Furthermore [Asmi&Ishaya,2012] uses the semantic value from [SentiWordNet], which does not contain Danish words. The conclusion is that basically [Asmi&Ishaya,2012] has only served as inspiration to our own approach to handle diminisher/boosters.

Our solution to the problem is to look for connective word classes. If the negation only affects words in the same sentence and if the sentence contains connective words then the scope is

1. from the start of the sentence to the connective word
2. from the connective word to the next connectivity word
3. from connective word to the end of the sentence

An example of this is the sentence

*The car does not have good driving ability **but** the design is excellent.*

In this example the negation will only have an effect before the "but" word, according to the mentioned scope rules in the above listing. If BOW was used then "the design" would also be negated.

However this scoping will not work in all cases, i.e. it will fail on recitation sentences, but we still believe that it will do better than simply having the scope on sentence level, similar to the BOW approach.
Another reason for not making the scope rules as specific as [Asmi&Ishaya,2012] is that we have identified a lot of noise in the evaluation texts. The evaluation texts contain many incorrect sentences, which may be a problem for the [Asmi&Ishaya,2012] technique, as it depends on finding noun and verb phrases.

The semantic value is changed differently for the different variants mentioned in section 4.1.1 "Variations". The "*ClusteringDanish"* and "*Distance Delta TFIDF*" are changed in the same way. When a negation is detected the distance weight is calculated by taking the longest distance. For example if the shortest path to good is distance 3 and the distance to bad is 7, then normally 3 words have been used, but when negation is detected the value 7 will be used.

The "*Delta TFIDF*" is changed by changing a positive number to a negative and vice versa in the opposite situation.

4.2.2 Diminisher/booster

The detection of a diminisher/booster is identified by looking for two words which are semantic bearing words and which are located next to each other. This is done to simplify the scope of the target for the diminisher/booster. In most cases it can be assumed that the target is the last word. Another advantage within this approach is that this makes the approach unsupervised learning. The semantic value of the words is found by using the distance graph. If a path can be found to the reference word then the word has a semantic value. Now we have identified the situation in which we have either a diminisher or a booster, but we need to establish which it is. Inspired by [Asmi&Ishaya,2012] a similar rule set table has been created. However, we disagree with some of the proposed rules which are presented in Table 8.

					Evaluation Votes	
Negation	**First word**	**Second Word**	**[Asmi&Ishaya,2012]**	**Our Conclusion**	**Negative**	**Positive**
True	Negative	Positive	Positive	Negative	4	1
False	Negative	Positive	Negative	Positive	1	4
True	Negative	Negative	Negative	Positive	1	4
False	Negative	Negative	Positive	Negative	5	0

Table 8 Rule set disagreement

To verify our conclusion the evaluation team received a test and they were asked to evaluate these combinations. In this test the evaluator just needed to determine whether the sentences were negative or positive. Our conclusion is supported by the evaluation team, but there are exciting sub results in this test. In some cases the evaluator actually wanted to classify the result as neutral even though the sentence contained high polarity words. Another unexpected result was that the negation word would not always negate the semantic value of the sentence. Our case was simplistic and more samples need to be done to achieve a clear picture of the behaviour of negation, but we have still chosen to use our conclusion instead of that of [Asmi&Ishaya,2012]. The rest of the rule set is used from [Asmi&Ishaya,2012] as we agree with the conclusion.

These conclusions help us to create a table which is the rule set for determining when we are dealing with a diminisher and when we are dealing with a booster. This is illustrated in Table 9

Negation	**First word**	**Second Word**	**Type**
True	Positive	Positive	Diminisher
False	Positive	Positive	Booster
True	Positive	Negative	Diminisher
False	Positive	Negative	Diminisher
True	Negative	Positive	Booster
False	Negative	Positive	Booster
True	Negative	Negative	Diminisher
False	Negative	Negative	Booster

Table 9 Rule set for determine type

In a situation where two subsequent semantic words are detected the first word is removed from the feature set and the second word's semantic value is changed depending on the assigned type. The "*ClusteringDanish"* and "*Distance Delta TFIDF*" use the same calculation. The distance metric is multiplied by 0.5 if it is defined as a booster and 1.5 if it is a diminisher. The reason is that a short path is equal to a higher semantic value and a longer path is equal to a lower semantic value.

The "*Delta TFIDF*" is changed by multiplying by 1.5 if it is defined as a booster and 0.5 if it is a diminisher.

5 Experiments

This section presents how we measure our experiments and which of the measurements we aim to improve and why. The ground truth is described; how it was created and why we are rerunning the corpus experiments. Finally the measurements are presented from the new experiments as well as our observations regarding the experiments.

5.1 Evaluation methodology

The following section is an extract from our preliminary work. Small changes have been incorporated in this section.

According to [Sebastiani, 1999] the evaluation of document classifiers is typically conducted experimentally, rather than analytically. In order to evaluate a system analytically there should be a formal specification of the problem that the system is trying to solve. By performing an experimental evaluation the focus is to measure the algorithm's ability to perform the classification correctly.

We have decided to perform the evaluation by experiments based on the arguments described in the above section. It has been identified throughout [Sebastiani, 1999] and [Prabowo&Thelwall, 2009] that the most frequently used measurements are *accuracy*, *precision*, *recall* and *F1*. Firstly there will be a short introduction to the notation and afterwards the four measurements will be introduced.

fp: indicates the number of negative labelled documents that were incorrectly classified as positive. Documents in this class are document which are wrongly classified for the class and thereby an unexpected result

fn: indicates the number of positive labelled documents that were incorrectly classified as negative. Documents in this class are document which are missing from the result

tp: indicates the number of positive labelled documents that were correctly classified as positive. This is the expected result

tn: indicates the number of negative labelled documents that were correctly classified as negative. Documents that are correctly missing from the class, so the documents are absent from the result

All of this can be seen in Table 10 Confusion table .

	Classified positive	Classified negative
Actual positive	***true positive (tp)*** Correct result	***false negative (fn)*** Missing result
Actual negative	***false positive (fp)*** Unexpected result	***true negative (tn)*** Correct absence of result

Table 10 Confusion table

5.1.1.1 Accuracy

The accuracy measurement indicates how well an algorithm has performed in recognizing the correct classification for both the positive (tp) and the negative (tn) items.

The formula is: $A = \frac{tp+tn}{tp+tn+fp+fn}$

5.1.1.2 Precision

The precision measurement evaluates the algorithms on how well the identified labelled documents actually belong to the class.

The formula is: $P = \frac{tp}{tp+fp}$

5.1.1.3 Recall

A recall measures the likelihood of whether a random document should be in a given class. A 100% recall is trivial to achieve because one could just return all the documents for any query. This is the reason why one also needs to measure the relevance of the documents returned.

The formula is: $R = \frac{tp}{tp+fn}$

5.1.1.4 F1

F1 is a combination metric of recall and precision. In the F1 version the weight of the numbers are equal, whereas other versions differ, such as F2 where recall has the most weight and $F_{0.5}$ where more emphasis is on precision.

The formula is: $F1 = 2 * \frac{precision * recall}{precision + recall}$

5.1.2 Selection of measurements

All of the above formulas represent a different view on how correctly the algorithms perform. To our knowledge no algorithm exists which excels in all of the measurements, even if there is a high likelihood of negative performance compared to a specialized algorithm for one of the measurements. This has lead us to understand when to use the different formulas to help determine which algorithm is best for the purpose the algorithm is going to be used in.

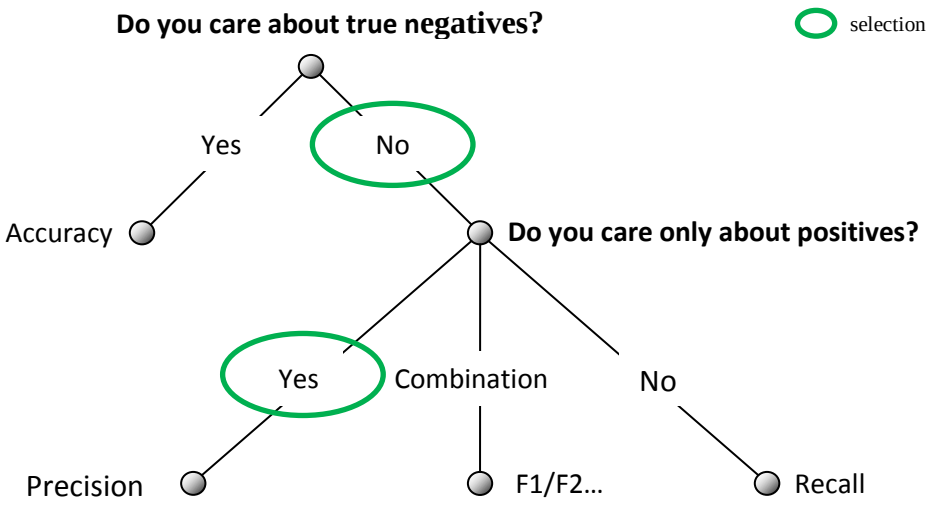


Figure 8 Decision tree for measurements

A decision tree (Figure 8) can help determine the measurement which suits ones needs. First one needs to determine if the usage situation demands that one cares about true negatives. In our case the usage situation is to determine if a certain post is positive, negative or neutral/not classified. In the Use Case we do not care about true negatives, which mean that accuracy is not the correct measurement for our situation. However accuracy is used a lot in the articles we have read for semantic analysis. These articles only work with 2 classifications; negative and positive. In this

situation, when the document is not correctly classified, then the only other class it can be a part of is the opposite, because semantic classification typically does not work with soft clustering. In this situation the accuracy measurements are interesting, but we have 3 classes of classification. The next question in the decision tree is “Do you care only about positives?” as in true positives with respect to true positives and true negatives. As we only care about correctly classified in our Use Case the measurement for us is precision. Precision will have the measurement which will determine the best algorithm for our purpose, but we will still calculate the other measurements to offer the community insight as well as for gaining more experience in the area. It is possible that we will need an algorithm which excels in one of the other measurements one day and by calculating it now we will understand how well these algorithms could be used in this new Use Case situation.

5.2 Rerun Corpus Experiment

One of the algorithms from our preliminary report had an advantage, namely the way the algorithm was trained. The algorithm was the corpus algorithm and it was the best performing algorithm from the preliminary report. The training was conducted on the same documents as the algorithm was evaluated on. However, the training source was on sentence level and the evaluation level was carried out on document level. This was done to save resources on manual classification.

The delta TFIDF algorithm needs training data to work. This time around we have chosen to train the algorithm in the standard 80%/20% method, see section “5.3 Ground truth”. We have chosen to take advantage of the new training data by reusing it for the corpus algorithm by rerunning the experiment. This will provide us with feedback on whether the corpus has an advantage in comparison with our preliminary report and/or confirm the validation of whether the previous training method is one that can be used to save time on creating training material.

5.3 Ground truth

We have collected data from [Amino] that we would like to have classified within the following three categories: positive, negative or not classified/neutral. In our previous work we had taken the role of being annotators by performing the classification ourselves. There are of course benefits and drawback to this approach, e.g. we were not dependent on people outside the group and thereby we could complete the task within reasonable time. However, we were probably influenced by the fact that we knew in advance how the annotated data would be used, which could have had an impact on

our classification, since we ourselves had developed and applied the methods they were to be tested against. By using external annotators we eliminate the possible pre-conceived classification.

In this project we have chosen a different approach and have assembled a group of annotators, more precisely five people. These annotators have different backgrounds; some of them work in IT, respectively, as project managers and system developers. Furthermore, there is a sociology student and a lawyer. To communicate with a group with different backgrounds can be a challenge and it certainly demands that we convey our messages accurately. The challenge for us was to transform data in a visual and understandable way and thereby minimize misunderstandings and make it easier for the annotators to perform the classification.

Based on our intuitive thinking and inspired by [Theresa Wilson, 2005] we prepared an annotation scheme. Figure 9 Snapshot annotation scheme" represented as an xml file was handed out to the respective annotators along with a short description in Text 1.

The Figure 9 Snapshot annotation scheme *illustrates a post obtained from the Internet that contains multiple sentences. Please provide a classification of the individual "sentence" as well as a classification of the entire post.*

The classification can be assigned in the following categories

- *1 - positive*
- *-1 - negative*
- *0 – not classified/neutral*

The classification is carried out by reading the contents of a "sentence" and relate to the three categories under which the variable "sematics" is assigned a value. Finally the variable "post semantic" is assigned a value. It is important here that it is the human intuition that determines not the number of phrases of the respective categories.

Text 1 Annotation guideline

```
<post sematic="0" postID="3301">
    <sentences sentence="Jo, det er stedet - så længe de ansatte fortsætter." sematics="0"/>
    <sentences sentence="Det er cvr nr. eller dig hvis det er et personligeget firma der har overenskomsten." sematics="0"/>
    <sentences sentence="Har du et link der kan underbygge din påstand?" sematics="0"/>
    <sentences sentence="Her er et link der underbygger min: http://www.startvaekst.dk/aftalelukning" sematics="0"/>
    <sentences sentence="Sammenlign mobilpriser med eller uden abonnement Amino-karma 1.008 " sematics="0"/>
</post>
```

Figure 9 Snapshot annotation scheme

The annotation schemes were collected from the five annotators and a comparison of the various posts was carried out. In situations where there were differences in the classification, it was decided by a majority vote. The result of the manual classification was a XML document, which was loaded back into the database. In this way all of the results are located in the database, which makes it easier to generate the measurements for the algorithm.

The ground truth dataset consists of 27 positive, 98 negative and 80 neutral posts.

5.4 Results

We have made three different categories of enhancements. We will investigate the result for each enhancement and at the end we will look at the overall result. We have chosen to display all the measurements; however as described in section "5.1.2 Selection of measurements" our interest is focussed on the precision measurement. The best scores are marked with green in the precision column along with an average score. The enhancement can be combined in many different ways. However, we have not tried every combination possible. In order to understand the results a naming convention has been introduced. The name consists of three elements. Each element is separated by a "/". The naming follows this pattern "*calculationName/number of clusters/ValanceShifters*". The first element is the name of the calculation which is used. The second element is the number of clusters that is generated and the last element indicates if the valance shifters are used. The valance shifter is abbreviated to VS and is only shown if enabled, if not the third element is removed. An example is "*ClusterDanish/2/VS*". This means the original cluster calculation is used, two clusters are generated and is improved with valance shifter. The rerun experiment of corpus is called "*new corpus*" and the old experiments are excluded from the results.

5.4.1 Calculation variants

In this section we will present how the different calculations scored against each other, which can be seen in Table 11.

	Positiv				Negative			
	Precision	Recall	Accuracy	f1	Precision	Recall	Accuracy	f1
ClusterDanish/2	0,0704	0,1852	0,5707	0,1020	0,5143	0,3673	0,5317	0,4286
Delta TFIDF/2	0,0000	0,0000	0,8683	0,0000	0,5816	0,8367	0,6341	0,6862
Dis+Delta /2	0,0870	0,0741	0,7756	0,0800	0,5593	0,6735	0,5902	0,6111

	Neutral				Average			
	Precision	Recall	Accuracy	f1	Precision	Recall	Accuracy	f1
ClusterDanish/2	0,6719	0,5375	0,7171	0,5972	0,4189	0,3633	0,6065	0,3759
Delta TFIDF/2	0,6719	0,5375	0,7171	0,5972	0,4178	0,4581	0,7398	0,4278
Dis+Delta /2	0,6719	0,5375	0,7171	0,5972	0,4394	0,4283	0,6943	0,4294

Table 11 Results for calculation variants

Since the algorithm only creates two clusters the category neutral/not classified is the same. All three algorithms use the same word list, as well as the word classes, to find the words of interest.

If we look at the precision score for positive the *"Delta TFIDF/2"* could not find a single positive document and generally the precision scores for positive are low. This indicates that the problem with identifying positive documents is still present, as it was also difficult for the preliminary report algorithms. The best algorithm for identifying positive documents is the *"Dis+Delta/2"*; the increase is minimal, but still an improvement of 23%. The result however is still not very usable.

The negative precision is 0,067 higher than the original cluster algorithm which is an improvement of 13%. The *"delta TFIDF"* is the component which improves the result, because the original version is the one with the highest improvement and when compared with the distance the effectiveness of identifying uniqueness of the n-gram is reduced, which makes it more difficult for the clustering algorithm to separate the documents into the clusters. This results in a lower precision.

The best average precision algorithm is the *"Dis+Delta /2"*. The reason for this result is found in the ability to find positive documents, which the "delta TFIDF/2" was not able to. The *"Dis+Delta/2"* is about 4.5% better than any of the other algorithms. Also the *"clusterDanish"* do not win any of the categories.

5.4.2 Number of Clusters

This experiment generates three clusters instead of two in order to improve the precision of the not classified/ neutral. The scores can be seen in Table 12. However, our changes resulted in a lower precision for this category. The negative category is the only one which is improved by creating an extra cluster, but the improvement is merely 0.5 %. This suggests that the not classified/ neutral is difficult to separate from the positive and negative for the clustering algorithm. The not classified/ neutral documents, which should be placed in the third cluster, contain words from the selected word class. However these word calculations are too close to the other documents, which make it difficult for the clustering algorithm to separate them into a separate class. The "Dis+*Delta/3*" is the one which performs best with a third cluster. In the positive the "Dis+*Delta/3*" performs equally well as does it with two clusters and it also has a slight improvement for negative, but for not classified/ neural the number is lower. Adding the third cluster did not have the effect on precision that we had hoped for.

	Positive				Negative			
	Precision	Recall	Accuracy	f1	Precision	Recall	Accuracy	f1
ClusterDanish/3	0,0000	0,0000	0,8634	0,0000	0,5846	0,7755	0,6293	0,6667
Delta TFIDF/3	0,0870	0,0741	0,7756	0,0800	0,5664	0,6531	0,5951	0,6066
Dis+Delta/3	0,0870	0,0741	0,7756	0,0800	0,5652	0,6633	0,5951	0,6103
ClusterDanish/2	0,0704	0,1852	0,5707	0,1020	0,5143	0,3673	0,5317	0,4286
Delta TFIDF/2	0,0000	0,0000	0,8683	0,0000	0,5816	0,8367	0,6341	0,6862
Delta+dis/2	0,0870	0,0741	0,7756	0,0800	0,5593	0,6735	0,5902	0,6111

	Neutral				Average			
	Precision	Recall	Accuracy	f1	Precision	Recall	Accuracy	f1
ClusterDanish/3	0,5811	0,5375	0,6683	0,5584	0,3886	0,4377	0,7203	0,4084
Delta TFIDF/3	0,6377	0,5500	0,7024	0,5906	0,4303	0,4257	0,6911	0,4257
Dis+Delta/3	0,6567	0,5500	0,7122	0,5986	0,4363	0,4291	0,6943	0,4297
ClusterDanish/2	0,6719	0,5375	0,7171	0,5972	0,4189	0,3633	0,6065	0,3759
Delta TFIDF/2	0,6719	0,5375	0,7171	0,5972	0,4178	0,4581	0,7398	0,4278
Delta+dis/2	0,6719	0,5375	0,7171	0,5972	0,4394	0,4283	0,6943	0,4294

Table 12 Results for experiments for the number of clusters

5.4.3 Valance Shifter feature

By adding the valance shifter an increase in precision for positive and negative documents is expected. Since the experiments are done with two clusters no change in not classified/ neutral is expected. The valance shifter has a high improvement in the positive class for "*ClusterDanish/2/VS*", which can be seen in Table 13. The improvement is 180% which is rather

substantial and the negative precision is also improved with 11.5%. The "*Delta TDIDF/2/vs*" also improves precision for the positive class, but this feature has a negative impact on the precision for the negative class. The valance shifter has no impact for the "Dis+*Delta/2/vs*" compared to "Dis+*Delta/2*". The valance shifter feature is a great success for the "clusterDanish" variant. This can be seen for the average precision score with an improvement of 15%.

	Positive				Negative			
	Precision	Recall	Accuracy	f1	Precision	Recall	Accuracy	f1
ClusterDanish/2/VS	0,2000	0,0370	0,8537	0,0625	0,5735	0,7959	0,6195	0,6667
Delta TFIDF/2/VS	0,1111	0,0741	0,8000	0,0889	0,5691	0,7143	0,6049	0,6335
Dis+Delta/2/VS	0,0870	0,0741	0,7756	0,0800	0,5593	0,6735	0,5902	0,6111
ClusterDanish/2	0,0704	0,1852	0,5707	0,1020	0,5143	0,3673	0,5317	0,4286
Delta TFIDF/2	0,0000	0,0000	0,8683	0,0000	0,5816	0,8367	0,6341	0,6862
Delta+dis/2	0,0870	0,0741	0,7756	0,0800	0,5593	0,6735	0,5902	0,6111
	Neutral				Average			
	Precision	Recall	Accuracy	f1	Precision	Recall	Accuracy	f1
ClusterDanish/2/VS	0,6719	0,5375	0,7171	0,5972	0,4818	0,4568	0,7301	0,4421
Delta TFIDF/2/VS	0,6719	0,5375	0,7171	0,5972	0,4507	0,4420	0,7073	0,4399
Dis+Delta/2/VS	0,6719	0,5375	0,7171	0,5972	0,4394	0,4283	0,6943	0,4294
ClusterDanish/2	0,6719	0,5375	0,7171	0,5972	0,4189	0,3633	0,6065	0,3759
Delta TFIDF/2	0,6719	0,5375	0,7171	0,5972	0,4178	0,4581	0,7398	0,4278
Delta+dis/2	0,6719	0,5375	0,7171	0,5972	0,4394	0,4283	0,6943	0,4294

Table 13 Results for valance shifter experiments

5.4.4 Overall

Now we compare the result with the preliminary report to find the best algorithm, which is presented in Table 14. Unfortunately none of the new algorithms perform best in any of the classes. There are two competitors; the SO algorithm based on "*google*" and our "*new Corpus/ 80%*". The "*new Corpus/80%*" is the algorithm which has the highest average precision. The best of the new algorithms is "*ClusterDanish/2/VS*" but for the average precision the "*ClusterDanish/2/VS*" is 21% lower in precision that the "*new Corpus/80%*".

	Positive				Negative			
	Precision	Recall	Accuracy	f1	Precision	Recall	Accuracy	f1
Bing	0,0000	0,0000	0,8537	0,0000	0,6090	0,8265	0,6634	0,5772
Google	0,1932	0,6296	0,6049	0,2957	0,7237	0,5612	0,6878	0,5620
new Corpus/100%	0,4000	0,0157	0,8644	0,0303	0,4615	0,0736	0,8252	0,8156
new Corpus/80%	0,6250	0,0394	0,8676	0,0741	0,4815	0,7975	0,8263	0,8177
new Corpus/60%	0,4091	0,3543	0,8443	0,3797	0,4367	0,4233	0,8061	0,7013
Naive	0,1613	0,1852	0,7538	0,1724	0,6023	0,5408	0,5897	0,6322
ClusterDanish/2	0,0704	0,1852	0,5707	0,1020	0,5143	0,3673	0,5317	0,4286
Delta TFIDF/2	0,0000	0,0000	0,8683	0,0000	0,5816	0,8367	0,6341	0,6862
Delta+dis/2	0,0870	0,0741	0,7756	0,0800	0,5593	0,6735	0,5902	0,6111
ClusterDanish/3	0,0000	0,0000	0,8634	0,0000	0,5846	0,7755	0,6293	0,6667
Delta TFIDF/3	0,0870	0,0741	0,7756	0,0800	0,5664	0,6531	0,5951	0,6066
Dis+Delta/3	0,0870	0,0741	0,7756	0,0800	0,5652	0,6633	0,5951	0,6103
ClusterDanish/2/VS	0,2000	0,0370	0,8537	0,0625	0,5735	0,7959	0,6195	0,6667
Delta TFIDF/2/VS	0,1111	0,0741	0,8000	0,0889	0,5691	0,7143	0,6049	0,6335
Dis+Delta/2/VS	0,0870	0,0741	0,7756	0,0800	0,5593	0,6735	0,5902	0,6111

	Neutral				Average			
	Precision	Recall	Accuracy	f1	Precision	Recall	Accuracy	f1
Bing	0,6232	0,5375	0,6927	0,5772	0,4107	0,4547	0,7366	0,3848
Google	0,8293	0,4250	0,7415	0,5620	0,5820	0,5386	0,6780	0,4732
new Corpus/100%	0,6999	0,9771	0,6939	0,8156	0,5205	0,3555	0,7945	0,5538
new Corpus/80%	0,7030	0,9771	0,6981	0,8177	0,6032	0,6047	0,7973	0,5698
new Corpus/60%	0,7840	0,8104	0,7140	0,7970	0,5433	0,5293	0,7881	0,6260
Naive	0,5395	0,5857	0,6718	0,5616	0,4343	0,4372	0,6718	0,4554
ClusterDanish/2	0,6719	0,5375	0,7171	0,5972	0,4189	0,3633	0,6065	0,3759
Delta TFIDF/2	0,6719	0,5375	0,7171	0,5972	0,4178	0,4581	0,7398	0,4278
Delta+dis/2	0,6719	0,5375	0,7171	0,5972	0,4394	0,4283	0,6943	0,4294
ClusterDanish/3	0,5811	0,5375	0,6683	0,5584	0,3886	0,4377	0,7203	0,4084
Delta TFIDF/3	0,6377	0,5500	0,7024	0,5906	0,4303	0,4257	0,6911	0,4257
Dis+Delta/3	0,6567	0,5500	0,7122	0,5986	0,4363	0,4291	0,6943	0,4297
ClusterDanish/2/VS	0,6719	0,5375	0,7171	0,5972	0,4818	0,4568	0,7301	0,4421
Delta TFIDF/2/VS	0,6719	0,5375	0,7171	0,5972	0,4507	0,4420	0,7073	0,4399
Dis+Delta/2/VS	0,6719	0,5375	0,7171	0,5972	0,4394	0,4283	0,6943	0,4294

Table 14 Overall result of all the experiments

Another striking result is that almost all of the algorithms are having trouble identifying positive documents compared to the negative class, since the precision score is so much higher for the negative classification. How can this be? We can exclude the idea of a bug in the implementation of the algorithm, because we have so many different implementations. The best algorithms for identifying positive documents are "*new Corpus*", "*Google*" and "*ClusterDanish/2/VS*". The

"*Google*" and the "*ClusterDanish/2/VS*" have a valance shifter feature included into the algorithm and "*new Corpus*" might indirectly also have valance shifter included as well. The reason for the indirect negation is that if the manually classified text is seeing ten instances of "not good" and two instance of "good", then the word will be negative even though we know the word to be positive, but because we do not detect the negation, the dictionary will conform to the norm of the domain. If the word "good" is mostly used with negation then, seen from a human perspective, the ratio contains negation. The "*new Corpus*" is the only one which has a lexicon that is generated from the same domain, meaning that the individual words can be affected by a negation in the documents. However, the other algorithms with valance shifter features did not improve the result, which suggest that the valance shifter is not the reason for the improvement.

We have a general problem with all the clustering experiments. The problem is the quality of the seed documents. We have not been able to identify documents which are rarely classified wrongly, which can be seen in Table 15. The column "quality" in Table 15 describes how many times one of the selected seed documents was correctly identified in the seed run, which consisted for 100 runs. "Num. of documents" describes how many documents were used from each of the classes as seed documents. The column "Num. of runs failed to classify" is the number of failed attempts to label the generated clusters. This happens because the labelling is performed by finding the cluster for the highest number of positive and negative seed documents. If the highest number for positive and negative seed documents is in the same cluster then it is not possible to label the clusters. This is the situation which the column represents. This also means that the variances can be high from run to run. The quality was very different from the different variances of the clustering algorithm. We have tried to keep the quality as high and as similar as possible, which is a compromise between number of documents and the seed document quality.

	Quality		Num. of documents				
	Highest	Lowest	positive	negative	Not classified/neutral	num. Of runs failed to classify	Average Precision
ClusterDanish/2	82	73	9	5	0	64	0,4189
Delta TFIDF/2	86	60	3	11	0	14	0,4178
Dis+Delta/2	92	67	5	11	0	14	0,4394
ClusterDanish/3	77	57	5	3	5	57	0,3886
Delta TFIDF/3	89	46	3	3	5	20	0,4303
Dis+Delta/3	89	63	3	5	5	17	0,4363
ClusterDanish/2/VS	100	24	5	5	0	22	0,4818
Delta TFIDF/2/VS	100	55	3	5	0	8	0,4507
Dis+Delta/2/VS	100	68	3	5	0	19	0,4394

Table 15 Seed documents quality

The highest average precision algorithm ("ClusterDanish/2/vs") is actually the one algorithm with the lowest seed document quality. The lowest quality documents are from the positive class, but "ClusterDanish/2/vs" is actually the one with the best precision in identifying positive documents and average with the number of failed runs to classify. This is counter intuitive, because with low quality the intuitive result would be a lower average and a high number of failed runs. We are not able to find a pattern which can explain the results we are seeing, except that this could be a signal that the clusters are very unstable even when run 20 times successfully.

This has led to another experiment where we have run "*ClusterDanish/2/vs*" ten times in a row to see how stable the results are. The precision scores can be seen in Figure 10.

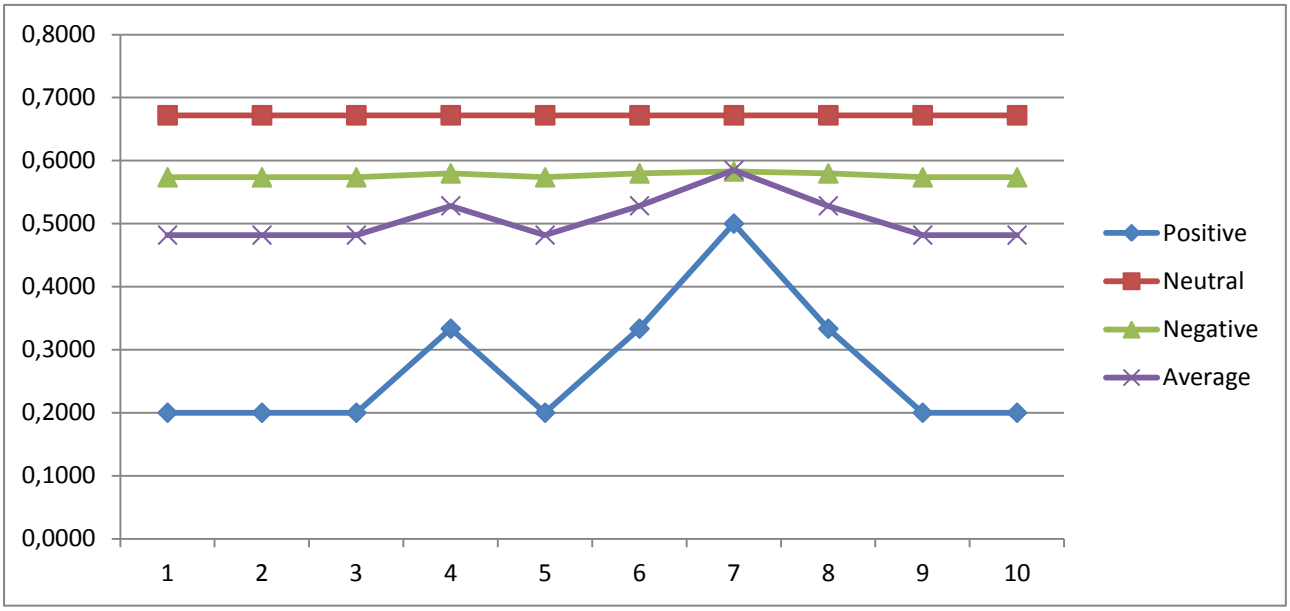


Figure 10 Ten runs of ClusterDanish/2/vs for precision

The average and negative score are reasonably stable, but the positive is unstable, which was also expected since the seed quality is low for the positive seed documents. However, even the best result would not change the overall conclusion, because the best score is not better than the "*new Corpus*" and "*Google*". This also proves that it is not enough to stabilize the clustering based algorithm with many runs. The seed documents also need to have the necessary quality to get a result.

5.4.5 Rerun Corpus Experiment

The result from the corpus rerun can be seen in Table 16.

	Positive					
	new Corpus/100%	Corpus/100%	new Corpus/80%	Corpus/80%	new Corpus/60%	Corpus/60%
Precision	0,4000	0,0000	0,6250	0,6000	0,4091	0,6667
Recall	0,0157	0,0000	0,0394	0,1111	0,3543	0,3704
Accuracy	0,8644	0,8683	0,8676	0,8732	0,8443	0,8927
f1	0,0303	0,0000	0,0741	0,1875	0,3797	0,4762

	Neutral					
	new Corpus/100%	Corpus/100%	new Corpus/80%	Corpus/80%	new Corpus/60%	Corpus/60%
Precision	0,6999	0,4020	0,7030	0,4675	0,7840	0,6635
Recall	0,9771	1,0000	0,9771	0,9875	0,8104	0,8625
Accuracy	0,6939	0,4195	0,6981	0,5561	0,7140	0,7756
f1	0,8156	0,5735	0,8177	0,6345	0,7970	0,7500

	Negative					
	new Corpus/100%	Corpus/100%	new Corpus/80%	Corpus/80%	new Corpus/60%	Corpus/60%
Precision	0,4615	1,0000	0,4815	0,9677	0,4367	0,8837
Recall	0,0736	0,0612	0,7975	0,3061	0,4233	0,7755
Accuracy	0,8252	0,5512	0,8263	0,6634	0,8061	0,8439
f1	0,1270	0,1154	0,1368	0,4651	0,4299	0,8261

Table 16 Rerun corpus experiments with different training data

The "*corpus/100%"* has shifted from not being able to classify positive documents to achieving 0.4 in precision; also neutral has gone up from 0.40 to 0.70. This improvement has had a negative effect on the classification of negative documents, which has fallen from 1 to 0.46. This is also the case for "*corpus 80%*" and almost the same for "*corpus 60%*". The difference is that precision dropped for "*corpus 60%*", whereas the others improved. The reason for this change can be found in the difference of the training data ratio of positive and negative documents. In the original training data the ratio was 21.6% positive and in the new training data the ratio was 60.6% of positive documents. This indicates that training and evaluating on the same data, on different levels, will hit the same ratio regarding positive/negative, which eliminates this parameter as a factor in the evaluation; however in real life this will be a factor. This means that it is not a good idea to train and evaluate on the same set of data, even if the level is different. This also shows that the corpus algorithm has an issue if the ratio in the training data and the evaluation data is not the same, because the difference between the scores in the two experiments is quite large.

We also stated that the corpus had optimum conditions for performing in the preliminary report. However the scores show that the corpus can perform better in some of the measurements when trained on different documents than those evaluated. This means that the argument with optimum conditions is untrue.

6 Discussions

This section will contain our reflection on the "Clustering method on sentiment analysis" algorithm and the "Delta TFIDF" in section "3.5 Algorithms for research". We analyse the pros and cons of each approach for performance, time consumption, implementation difficultness, concept and sources of the semantic value e.g. wordlist.

6.1 Clustering method on sentiment analysis algorithm

The "clustering method on the sentiment analysis" algorithm is the base for our implementations. This algorithm is more difficult to implement than any of the algorithms from the previous work. There are many concepts in this approach which at first is easy to understand and very intuitive. Especially the distance idea, where a word's semantic value is determined by the distance from the word to the reference words, where the reference words are the most positive and negative words. The distances seem conceptually correct from the mathematically perspective that the closer the word is to the reference word the more semantic value they share. Also this approach is very modular in its construction, which makes it easy to change or add features.

The challenging parts are to calculate the distances to the reference words, clustering execution, POS tagging of the words and finding the seed documents.

We have previously used a POS tagger, which in this project reduced the difficulties as well as the implementation time. The difficulties of the clustering is mainly a technical issue in regards to not knowing any of the implementations available, but also relating to how to fine tune the parameters for the clustering mechanism to archive the best results. These two issues are common issues and are not specific to the algorithm as such.

However, the distance measurement and the seed documents are challenges which relates from the algorithm. The two challenges are dependent on the language used to analyse on and the documents being analysed. If the language is English, or one of the larger languages, then the distance measure is easier because it is possible to find an API, such as [yago-naga], which can give you the distances. However, when running the analysis on a small language such as Danish no such resource is available that we know of. If such a resource did exist one had to generate a graph from that source and then implement a graph calculation to find the definitions. Typically graph based implementation is more difficult because of the possibility of infinity loops. In our previous work

the challenges was to find lexicon sources. However in our experience the synonym lexicon source is easier to find than the previous lexicon sources which we needed, which was a list of positive and negative words.

The other challenge with the algorithm is to get hold of seed documents. The authors [Li&Liu, 2012] recommend finding the seed documents from labelled documents instead of generating the documents. The approach needs labelled data for finding seed documents. However, the first time we read the article we did not get the impression that labelled data was needed as the article gives the impression of being unsupervised. This is also true for the algorithm part as it only needs seed documents as input, the best approach demands labelled data to find the seed document. This time consumption is reduced if the evaluation level is at document level, and then it is only the document that needs to be evaluated. It is more time consuming annotating a document with the semantic annotations to generate a dictionary than just evaluating a document. The time consumption of this technique is higher than just evaluating the documents, since it is not certain that the documents evaluated will be semantically strong enough to become a seed document. A seed document needs to be evaluated correctly 100 out of 100 times. One cannot know in advance which documents will provide this result, meaning that one can evaluate many documents without finding a seed document. Our experience demonstrated a challenge and even more so if we want to reuse our evaluation / training data from our previous work, because the current data set may not contain seed documents. The challenges with finding seed documents, and the problems arising when the quality is not as high as possible, can be read in section 5.4 Results

The authors' [Li&Liu, 2012] approach for generating the seed documents was rather simplistic. The possibility of achieving a better result by applying a more complicated approach may solve this challenge of the algorithm.

If the seed documents are poorly selected then the clusters generated cannot be labelled. The "Clustering method on sentiment analysis" algorithm does not handle this issue- A simple solution to this problem is not to use the cluster execution and continue to create cluster runs until one has the number of cluster runs specified by [Li&Liu, 2012]. However, this makes the implementation less efficient. The algorithm is actually very quick (clock time) compared to the Semantic Orientation algorithm in the preliminary work. With the modular construction this algorithm is actually easy to parallel. For example the cluster runs can be run concurrently on different machines

with the help of [hadoop] with minimal effort to implement. However [hadoop] can be difficult to setup. Another disadvantage with this approach is the cost of the machines. Our approach supports multiple machines but it can also be run on just one. However the need for more machines is reduced the better quality your seed document is. This means that the cost can either go to more machine power or to improve the quality of the seed documents.

6.2 Delta TFIDF

The delta TFIDF is very easy to implement compared to the cluster algorithm. Delta TFIDF is not dependent on a source or API, which makes it language neutral. This however comes with a price, i.e. training data is needed. However, this time we know in advance that the documents that are being labelled can be used, unlike the situation for training data in the clustering approach. The performance should be very good, because of the simplicity of the algorithm and because the divide and conquer pattern can be used to scale the solution to multiple machines or threads.

A disadvantage of this algorithm is that it is very difficult to enhance or change, because one needs to fit the new change in to the scale which the algorithm works with. In the current version the delta TFIDF only evaluates n-gram, which means that an aggregation function is needed to evaluate on different levels such as sentence or document level. The aggregation function could for example be average. The advantage would be a reduction of the weight for high level semantic value in just one sentence and also give a more correct picture of the complete text.

7 Future work

This section will present our thoughts about the future work. Firstly, we will look at possible approaches in a big scale and secondly, we will address areas within this master thesis where we think that there could be interesting challenges or details to improve and test.

We have worked within the area of sentimental analysis performed on the Danish language for two semesters. During this work we have introduced several algorithms and experiments ranging from the naïve approach, counting positive and negative words, to more sophisticated methods e.g. the cluster approach. It might be interesting to introduce a framework where the different algorithms should interact in a chain. If one algorithm fails to classify a text it should be propagated to another algorithm in the chain. The idea is to improve the precision on those documents which are positive or negative but is misclassified as neutral by one of the algorithms.

We have a challenge when working with dictionary based algorithms. To the best of our knowledge the existence of sources in the Danish language is rather sparse and thereby restricting the use of these types of algorithms. However, a possible solution could be to translate texts in Danish to English in order to take advantage of the fact that the English language is far better supported within semantic analysis than the Danish language is. This approach is not without issues; it will introduce new challenges. Will the meaning be lost in the translation? Is it possible to maintain sarcasm and other characteristics of the Danish language? Another and far more time consuming task would be to begin to annotate texts ourselves, which could be used to create dictionaries on.

The k-means clustering algorithm is selecting the centroid randomly. One way of altering this approach of selecting candidates could be to use the documents that contain the most semantic words or choose a random document from the seed documents. The advantage of using a seed document is that the classification is known in advance and is correct. The idea is to reduce the time for the clustering to converge to the final result and perhaps stabilize the results.

As discussed in section “6.1 Clustering method on sentiment analysis algorithm”, we have a challenge finding good quality seed documents. We have found another article [Zagibalov&Carroll, 2008] with a different approach to generate the seed documents. If this approach could be used and generate good results, the challenge of finding good quality seed documents would be solved. At least the article could be used as inspiration in solving or improving on this issue. With auto

generated seed documents the time consuming task of classifying candidates for seed documents would be gone.

8 Conclusion

Our experiments illustrate that the biggest disadvantage is the process of finding seed documents for the clustering approach. The problem is that one does not know how many documents one needs to manually classify to achieve the high quality number of seed documents needed. This can be a time consuming process as one may have to manually classify a high number of documents before identifying the seed documents. This time could be used to annotate documents for the corpus algorithm and according to our experiments the corpus algorithm retrieved a higher average precision than the cluster based experiments.

In an early phase of our work, we decided that we would involve external people to classify our data. The reasoning for the decision is that we will minimize the impact that we could carry on data since we know how the data will be used later on in the test phase. We were able to assemble people with different educational backgrounds and age since diversity was important for us. When we contacted people to verify whether they wanted to participate in the project we only got positive feedback. However, it turned out that we had to spend a lot more time than anticipated to motivate and follow up on the process.

It is debatable whether it was a good idea to involve people from outside due to the time we had to spend managing the small group of people. One thing is certain, we can assure that we had no influence on how our annotators classified the data and thus we have removed the uncertainty about our impact on the test data.

The highest precision was achieved with the “ClusterDanish/2/VS” algorithm. This was due to the added feature of valance shifters, because the other combination with three clusters and without valance shifters was lower. The simple BOW worked well combined with the diminisher and booster and showed that even a simple implementation of valance shifters can make an improvement when the document size is small like in our example.

Even with all the improvement made to the clustering algorithm it was not possible to generate a better result than the “corpus” algorithms. Even the “Google” algorithm does perform better than the cluster based algorithm in our experiments. If we base the decision solely on the measurements the “corpus” is the winner, but it is also time consuming and domain depended. The issue with the “Google” algorithm is the economy of sending request to the Google search engine. The issue with

the clusters based algorithm is the process of finding the seed documents with high quality. All of the experiments for the clustering based approach are carried out on low quality seed documents. This is supported by the experiments where we run the “ClusterDanish/2/VS” for ten times and the fluctuation was great. The process of finding seed documents could perhaps be improved and there is also the possibility that another clustering method could improve the end result. The corpus algorithm is difficult to improve on precision and time consumption. The “Google” algorithm’s main issue is the economy; however this could be reduced by running a web scraper on the Danish web to generate the statistic material instead of requesting Google, which could be run on the same machine as the SA to reduce costs. There is no clear winner when looking at more factors that just the measurement precision.

9 References

9.1 Articles

[Vinodhini & Chandrasekaran, 2012]: G. Vinodhini & RM. Chandrasekaran, 2012. Sentiment Analysis and Opinion Mining: A Survey

[Pang, Lee & Vaithyanathan]: Bo Pang, Lillian Lee & Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques

[Rui Xia, 2011]: Rui Xia, Chengqing Zon & Shoushan Li, "Ensemble of feature sets and classification algorithms for sentiment classification"

[Turney,2002]: Peter D Turney, (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews

[Hatzivassiloglou&McKeown,1997]: Hatzivassiloglou & McKeown,1997. Predicting the semantic orientation of adjectives

[Li&Liu,2012]: Gang Li and Fei Liu,2012 . Application of a clustering method on sentiment analysis

[Gamon, 2004]: Michael Gamon, 2004. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis

[Theresa Wilson, 2005]: Theresa Wilson, Janyce Wiebe & Paul Hoffmann . Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis

[Asmi&Ishaya,2012]: Amna Asmi and Tanko Ishaya,2012 . Negation identification and calculation in sentiment analysis

[Wiegand et al.,2010]: Michael Wiegand, Alexandra Balahur, Benjamin Roth, Dietrich Klakow, AndrésMontoyo,2010. A survey on the role of negation in sentiment analysis

[Martineau&Finin, 2009]: Justin Martineau, and Tim Finin, 2009. Delta TFIDF: An improved Feature Space for Sentiment Analysis.

[Zagibalov&Carroll, 2008]: Taras Zagibalov & John Carroll, 2008. Automatic Seed Word Selection for Unsupervised Sentiment Classification of Chinese text.

9.2 Web resources

- [Amino]: http://Amino.dk

- [Wordnet]: http://wordnet.princeton.edu/
- [Dannet]: http://wordnet.dk/
- [LuceneStemming]: http://snowball.tartarus.org/algorithms/danish/stemmer.html
- [yago-naga]: http://www.mpi-inf.mpg.de/yago-naga
- [ParoleDefinition]: http://korpus.dsl.dk/paroledoc_dk.pdf
- [SentiWordNet]: http://sentiwordnet.isti.cnr.it/
- [OpenNLP]: http://opennlp.apache.org/

9.3 Books

[Zhu & Goldberg, 2009]: Xiaojin Zhu and Andrew B. Goldberg. Introduction to Semi-Supervised Learning Morgan & Claypool Publishers. ISBN: 9781598295481

9.4 Report

[Andersen & Hansen, 2012] Kim Andersen and Jesper Hansen. TAX – Sentiment analysis, 2012 Aalborg University.