Generation of a compressed and high quality information profile

Priya A. Kanhai

Internship report

VU University Amsterdam
Faculty of Sciences
Business Mathematics and Informatics
De Boelelaan 1081a
1081 HV Amsterdam

ParaBotS B.V.
Singel 160
1015 AH Amsterdam

December 2008

Preface

In order to complete the study Business Mathematics and Informatics a six-month internship needs to take place. I performed my internship at ParaBotS B.V. The assignment was to find a method that is able to automatically create compressed and high quality information profiles from large collections of documents. These profiles contain words that are characteristic for an entity, e.g., a concrete person.

I would like to thank Marten den Uyl for giving me the opportunity to perform my internship at ParaBotS and also for the many creative ideas he provided me during this period. Also thanks go to Stijn Prompers who has been my supervisor. I am grateful for his guidance, support, and the useful comments he provided me during my research at ParaBotS and in writing this thesis. My special thanks go to my supervisor at the VU, Wojtek Kowalczyk. I am grateful for his brilliant insight regarding my research problem. Without him, I would have definitely been headed into the wrong direction. I would also like to thank him for his guidance and the useful comments on my thesis.

Furthermore I want to thank everyone at ParaBots, VicarVision, and Sentient for their support during my research.

Last but not least, I would like to thank Sandjai Bhulai for being my second reader and all my friends and relatives who helped me with the selection of words.

Amsterdam, December 2008.

Management Summary

The ParaBotS Vox-Pop application lists each 3 hours the top 10 entities that are most talked about that day. Some of these entities in the top might be not known by the user. So, it would be interesting to have a tool that can generate a few words that are most characteristic for these entities. The goal of this project is to find a (mathematical) technique that is able to describe in a few words an entity, i.e., generate a profile of an entity. The profile should not only be short, but also representative, distinctive, and relevant. We regard this problem as a two-class classification problem. Documents that are related to an entity form a class “positive”, while documents that are related to other entities are “negative”. The following (feature selection) techniques are applied for this purpose: Oddsratio, Information Gain, Ripper, Relief, SVM as feature selection technique, and BoosTexter. We did not only consider single words, but also pairs of consecutive words, and lists that consist of both single and pairs of consecutive words (composed words). It is not only interesting to see which technique was able to generate a high quality profile, but also to look at the stability of the technique, i.e., which technique would generate the same profile given another set of documents from the “negative” class.

To measure the quality of the selected words we decided to look at the F_1-measure (for distinctiveness and representativeness) and the correlation between the words selected by humans and the words selected by our techniques (for relevance and representativeness). The stability of a technique was measured by the nominal concordance.

There was no significant difference between the different techniques when looking at the F_1-measure single words and composed words. However, for pairs of consecutive words there was a difference between our techniques. This difference was caused by Relief. Leaving this technique out, resulted in no significant difference between the rest of the 5 techniques for the F_1-measure.

The correlations between solutions made by humans and our methods were relatively weak. BoosTexter, Relief, and Information Gain yielded the best significant positive correlation for composed words. For both BoosTexter and Relief there were 6 of the 12 entities that showed a significant positive Kendall’s correlation coefficient. There were 4 out of the 12 entities that had a significant positive correlation coeffient between words selected by Information Gain and those selected by humans. Since, BoosTexter and Relief are both performing in the same way, we can look at other criteria for selecting either one of the two. BoosTexter is preferred above Relief when taking the CPU time into account. There is no clear choice between BoosTexter and Information Gain. The former performs slightly better than the latter, but it takes up to a couple of minutes to select the words, when the dataset is large, while the latter takes only a few seconds.

The Oddsratio turned out to be the most stable technique for single, pairs of consecutive, and composed words.

Table of Contents

1 Introduction

1.1 ParaBotS

ParaBotS develops applications and services that make sense of the ever growing stream of information on the Internet, in newspapers, and other media. State-of-the-art natural language processing and machine learning techniques are used for information retrieval, text mining, and web searching. ParaBotS was founded in 2001 and is located in Amsterdam.

1.2 Objective

One typically uses search engines, like Google, to find information about something. In most of the cases these search engines return a lot of information. For example, typing in Google "Geert Wilders" gives a list of 1,820,000 documents[1]. Going through all those 2 million documents would require a lot of time, but what if there was a tool that would describe in a few words this person "Geert Wilders", i.e., that would produce a short profile of "Geert Wilders"? This would be very useful and would save us a lot of time.

Currently, ParaBotS are extracting information from the Internet about entities in different categories. A category is a group of entities belonging together, where an entity can be a person, or a company, or a product. The extracted information, stored in a SQL database (called the Vox-Pop database), consists of many text documents. ParaBotS have an internally developed tool that can determine whether a document contains (positive or negative) information about an entity. Based on this information it determines which 5 entities are discussed frequently on the Internet. These entities are then listed on the site www.vox-pop.nl. However, it could be that there are one or more entities in the list that are not (widely) known, i.e., that when looking at the entity's name we think "Who is that?" The idea here is to generate a short profile for the entities such that this profile describes these entities in a few words. In other words, we need to build a tool that can produce a profile of an entity (given the documents in which the entity is mentioned). The goal of this internship is to create such a tool (or a prototype of it) that can describe in a few words an entity given the documents in the Vox-Pop database. These few words should be representative, distinctive, and relevant. The profile should contain few words, because it should be readable by humans. These words should also be representative, i.e., the list of words should provide a good representation of the entity. The third criterion states that the words should be distinctive. This means that these words should not apply to other entities in the same category. Last but not least, the words should be relevant, i.e., meaningful. These words are also called features or attributes in data / text mining.

[1] 1.820.000 voor "Geert Wilders" (0,23 seconden) on 14 October 2008

1.3 Problem statement

The main research problem that is addressed in this paper concerns a design of finding one or more methods that are able to produce compressed and high quality information profiles about entities given some documents. These methods should be implemented and evaluated.

The main research question here becomes:

Which (mathematical) technique(s) can be used to produce a profile of an entity such that this profile consists of a few representative, distinctive, and relevant words?

Different data mining techniques will be considered to answer the main research question.

1.4 Structure of the report

This report is organized as follows: Chapter 2 contains a (very) short introduction to text categorization and feature selection / construction. In Chapter 3 the feature selection techniques that will be used during this project are discussed. In Chapter 4 the implementation of the methods is presented. In Chapter 5 the data that is used is specified. Chapter 6 explains what evaluation technique and measures will be applied. Chapter 7 contains the experimental set-up that is used. In Chapter 8 the results are provided. The last chapter contains some conclusions and recommendations.

2 Background

In the last few years feature selection techniques have been applied for many reasons such as saving computational time and storage. These techniques are mostly applied in a text categorization context as the amount of information on the web is systematically increasing. We will apply feature selection methods to generate a profile for an entity. Once this profile has been produced, one needs to evaluate it. One machine learning technique that has been applied to text categorization will be used to evaluate this profile. The next two sections provide an introduction to text categorization and feature selection techniques.

2.1 Text categorization

The amount of information available on the Internet has grown exponentially in the past few years. Also the number of people putting text on-line, and using those texts has increased. Text categorization can help to order and organize this information **[13]**. Text categorization, also called text classification, is a process of automatically assigning documents to one or more predefined categories or classes based on their contents **[6; 9; 10; 14]**. Multiclass and multilabel are the two words that usually pop up in this context. We define a classification problem as multiclass in case there are more than two classes defined. Multilabel means that a document can belong to more than one category. Text categorization is a supervised task, i.e., the document labels / categories are provided. (In unsupervised learning these document labels are not given.) Machine learning techniques, such as k-Nearest Neighbor **[32]**, Naïve Bayes **[27]**, Decision Trees **[31]**, Neural Networks **[30]**, Support Vector Machines **[28; 29]**, Boosting **[2]**, Distributional Clustering **[13]**, have been applied to text classification problems in recent years.

In many text categorization problems a text is represented as a "bag of words" (BOW) **[15; 27; 31]**. This means that the text is transformed into a set of words and the word order is ignored **[15; 27; 31]**. In a BOW model one looks if the word is present or absent in a text and thus ignoring the word frequency **[14; 31]**. A BOW model is also called a unigram representation and it leads to a vector space model **[14]**.

One of the problems of text categorization is the high dimensionality of the feature space **[9]**. Imagine that a text in a document contains 50 unique words and that we have 100 documents where each document contains words that do not appears in any of the other remaining 99 documents. So, we obtain 5000 (unique) words in total. The feature space is now a vector with dimension 5000. Similarily, considering our example from Chapter 1 of "Geert Wilders" where there are almost 2 million relevant documents; it can lead to a feature space of dimensionality 1[1] million. It would require not only a lot of space, but also a lot of time to categorize these documents. In order to automatically reduce the space complexity and computational time, feature selection and / or construction is applied. Feature selection and / or construction will be discussed in the next subsection.

[1] Suppose we are using a English dictionary, then there are 988,968 words **[43]**

2.2 Feature Selection / Construction

Feature selection techniques have been applied to save storage, network bandwidth and computational time **[12; 16]**. Also, features obtained by feature selection can sometimes improve the classification accuracy **[12]**. In text categorization feature selection is mostly used for saving the computational time and to achieve high classification accuracy **[2; 11]**. Reducing the number of features can save a lot of computational time, while reducing the noise from the data can lead to an improvement of the accuracy **[2]**.

Feature selection, also called feature reduction, can be defined as the process of selecting a best subset of features (e.g., words in text categorization) from the original features that are relevant to the target concept **[14; 15; 18]**. Next to feature selection, we have feature generation. This is a process of generating new features from the original features and is called feature extraction, or feature construction **[14; 15]**. For this project we will focus on feature selection methods, and not feature construction methods. For feature construction method one needs to have a lot of knowledge about the features beforehand, which makes it less attractive to use. An example of a feature construction is that if features such as 'age=16', 'age between 13 and 18', 'position in the family= residential child' appear in a dataset then they can or should be labeled as 'young'. So, we need to know beforehand what can be labeled as 'young'.

There are many feature selection methods discussed in the literature for supervised learning. In **[17]** a Genetic Algorithm to select a feature subset is specified. Odds ratio, Document frequency, Information Gain, Mutual Information, a χ^2 statistic, and Term strength are also used as feature selection methods in **[2; 6; 9; 12; 14; 22; 26]**. In **[20]** a correlation Based Filter Approach to select a subset of features is presented. The Gini index is applied as a feature selection technique in **[21]**. Optimal Orthogonal Centroid Feature Selection for Text Categorization is a new feature selection technique that is introduced in **[22]**. In **[23]** a novel feature selection method that is based on mutual correlation is proposed. BoosTexter, a boosting-based system for text categorization, is explained in **[1]**. Feature selection methods can mostly be distinguished into two groups: filter and wrapper methods **[16; 17; 20; 33; 34; 35]**. The filter method operates independently of the learning algorithm, where the wrapper method uses the learning algorithm to select the features **[16; 17; 20; 33; 34; 35]** (see Figure 1). The results achieved by the wrapper method are often better than the ones obtained by the filter methods **[20; 33]**. However, the wrapper method is computationally very expensive compared to the filter methods and also causes overfitting **[17; 33; 35]**. Filter methods are able to scale large datasets better than wrapper methods **[35]**.

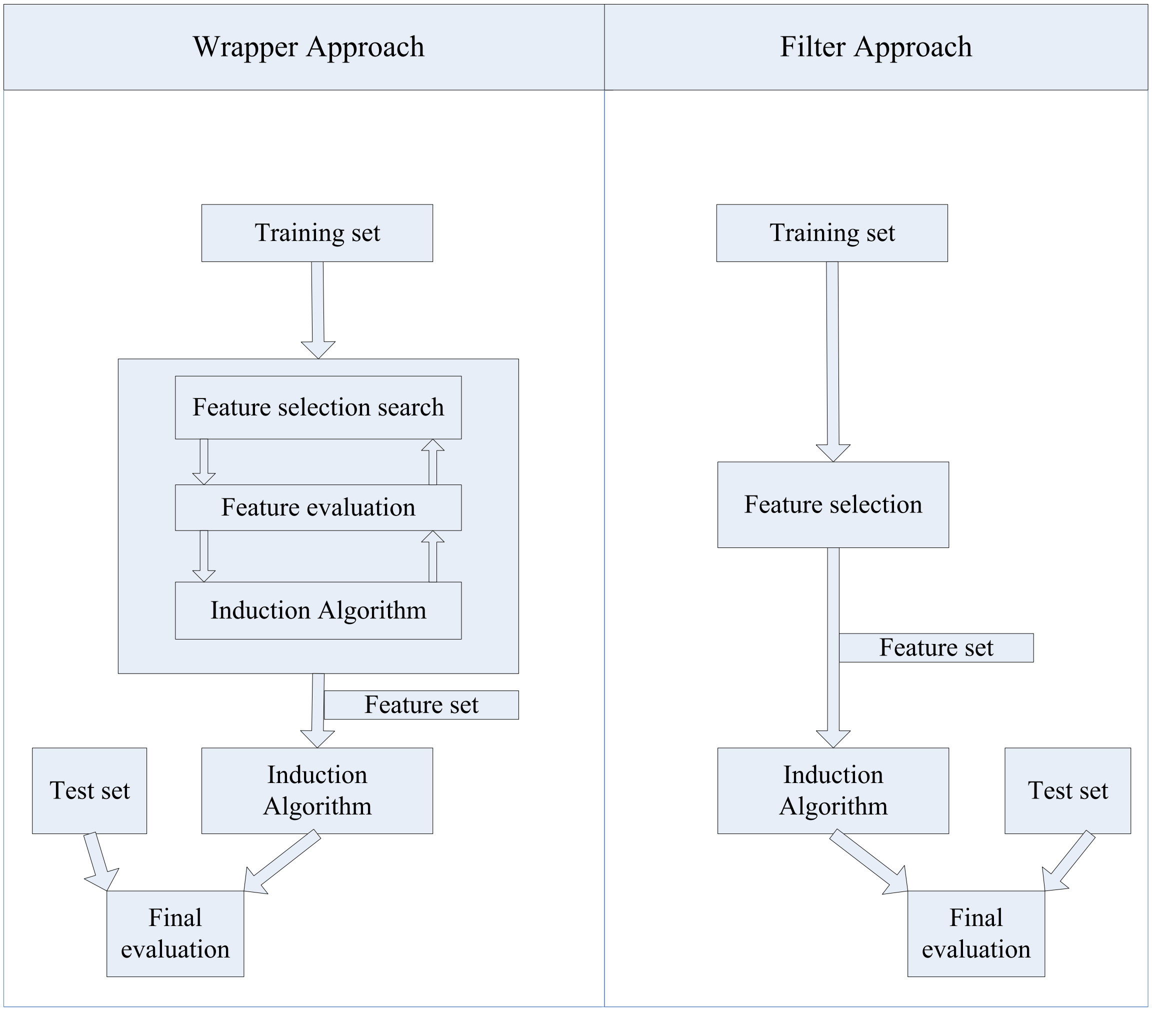

Figure 1: Global scheme of the wrapper and filter approach

As discussed earlier, feature selection methods are used to reduce the space complexity and / or to reduce the computational time. However, we will not use feature selection methods for these reasons. Existing feature selection methods will be applied to generate an entity profile. In general, feature selection algorithms are selecting those features that are able to distinguish between the positive and negative class, meaning these methods select distinct and also representative features. For this project feature selection methods will be applied to produce an entity profile that contains few, representative, distinct, and relevant words. From the different existing feature selection techniques that will be explored one technique will be chosen in the end. This technique should not only produce representative and distinct words, but also relevant words. The next chapter will discuss the feature selection techniques that will be considered during this project.

3 Feature Selection Techniques

There are many feature selection methods available in the literature. However, it is impossible to apply all these methods when considering the time available for this project. The objective is to select those methods that are most distinctive from each other. It was decided to use the following feature selection criteria: Odds ratio and Information Gain. The feature selection algorithms Ripper, Relief, SVM, and BoosTexter are also applied.

The Odds ratio algorithm makes use of the probability theory to select features. The central idea behind this algorithm is that the features in the relevant documents have a different distribution compared to those in the non-relevant documents **[11]**. Information Gain is another feature selection method that will be used during this project. This method uses the information theory rather than probability theory to select features. Information Gain is also a very popular **[22]** and a widely applied algorithm **[2; 6; 9; 11; 12; 26]**. In **[9]** it was found that Information Gain was one of the most effective measures for reducing the dimensionality. Other feature selection algorithms that will be applied are Relief and SVM. Relief uses a distance measure when selecting the features while SVM is able to reduce the feature size at the same time as maintaining a high accuracy **[26]**. Ripper is one of the algorithms that is also applied during this project. Ripper combines short models that have a high accuracy in a logical way. The short models are represented in terms of features that are combined in such a way that a high accuracy is obtained. The last algorithm that is used is BoosTexter. BoosTexter joins classifiers (e.g., rules) in order to obtain a final classifier that gives the highest performance. All these algorithms can be considered as filter methods. However, for BoosTexter it is a bit unclear whether it is a filter or a wrapper method.

The following notations will be used:

- D – domain of documents; $D = \{d_1, ..., d_z\}$
  - d is an arbitrary document
  - d_i is document i
- C – classes; $C = \{C_+, C_-\}$
  - C_+ is positive class, C_- is negative class,
  - c_i is class label of document i
- V – words; $V = \{v_1, ..., v_m\}$
  - v is an arbitrary word
  - v_i is word i
- W – weights
  - W[v] weight of word v

In the next subsections a detailed (mathematical) explanation of the six feature selection methods will be provided.

3.1 Odds ratio

As discussed earlier, Odds ratio uses probability theory to select the features. It assigns a high score to words that are characteristic for the positive class and a low score to those that are characteristic for the negative class. However, we could also get rare words instead of characteristic words for the positive documents **[26]**. This will happen when words occur only in a few of positive documents and not in the negative ones. The formula to calculate the odds ratio is provided in Figure 2.

$$\text{Odds ratio}(v) = \ln\left(\frac{P(v \mid C_+)[1 - P(v \mid C_-)]}{P(v \mid C_-)[1 - P(v \mid C_+)]}\right)$$

where

$P(v \mid C)$ is the conditional probability that is calculated as follows: the number of documents that belong to class C and contain word v divided by the total number of documents that belong to class C: $\frac{\#\text{documents}(v \mid C)}{\#\text{documents}(C)}$.

Figure 2: The Odds ratio algorithm

Suppose that we have 100 documents belonging to class C_+ and 500 documents belonging to class C_- where word v appears in 80 of the 100 positive documents and in 200 of the 500 negative documents. The odds ratio is then

$$\text{Odds ratio}(v) = \ln\left(\frac{80/100[1 - 200/500]}{200/500[1 - 80/100]}\right) = 0.78$$

3.2 Information gain

Information Gain (IG) determines how much information is gained about a class by taking into account the presence and absence of a word. This method uses the information gain to determine the importance of the feature. The calculation of the IG is shown in Figure 3.

$$IG(D, v) = \text{Entropy}(D) - \text{Entropy}(D \mid v)$$

$$\text{Entropy}(D) = \sum_{c \in C} - p(c) \log_2 p(c),$$

where

$p(c)$ can be seen as the proportion of documents D that belong to class $c \in C$

Figure 3: The IG algorithm

According to **[9]** the time complexity of computing the Entropy is $O(\#\text{words} * \#\text{classes})$. We will illustrate the calculation of the IG and the Entropy with an example. Suppose we have a set of 14 documents and we want to know the IG for word v. If we assume that Table 1 represents the values of word v, then the IG for word v is calculated as follows:

Documents d	Word v	Classes C
1	0	-
2	1	-
3	0	+
4	0	+
5	0	+
6	1	-
7	1	+
8	0	-
9	0	+
10	0	+
11	1	+
12	1	+
13	0	+
14	1	-

Table 1: Values of word v

There are 9 documents that belong to class C_+ and 5 documents that belong to class C_-. For word v, there are 8 documents where v = 0 and 6 documents where v = 1. Of these 8 documents with v = 0, 6 of these have class C_+ and 2 have class C_-. From the 6 documents with v = 1, there are 3 of these have class C_+ and 3 have class C_-.

The IG is calculated as:

$$IG(D, v) = Entropy(D) - Entropy(D \mid v)$$

$$IG(D, v) = Entropy(D) - \frac{8}{14} * Entropy(D_{v=0}) - \frac{6}{14} * Entropy(D_{v=1})$$

$$IG(D, v) = 0.940 - \left(\frac{8}{14} * 0.811\right) - \left(\frac{6}{14} * 1\right) = 0.048$$

where

$$Entropy(D) = \sum_{c \in C} - p(c) \log_2 p(c) = -\frac{9}{14} \log_2 (\frac{9}{14}) - \frac{5}{14} \log_2 (\frac{5}{14}) = 0.940$$

$$Entropy(D_{v=0}) = -\frac{6}{8} \log_2 (\frac{6}{8}) - \frac{2}{8} \log_2 (\frac{2}{8}) = 0.811$$

$$Entropy(D_{v=1}) = -\frac{3}{6} \log_2 (\frac{3}{6}) - \frac{3}{6} \log_2 (\frac{3}{6}) = 1.000$$

Thus the IG for word v in this example is 0.048.

3.3 Relief

The Relief algorithm is based on the distance measure. It searches for nearest hit and nearest miss given one or more randomly selected documents. Let us consider one randomly selected document R. Let us find a document that is closest to R and belongs to the same class as R (a "Hit"); and similarily another closest document that belong to the opposite class (a "Miss"). Then for every word v, Relief calculates the difference between the values of these two documents. If the word v appears in both documents, we say that the values of these documents are the same and the difference is equal to 0. This holds also if the word v does not appear in either of the two documents. In case the word v appears in one of the documents but not in the other one, we say that the values of these documents are the different and the difference is equal to 1. It sounds logical, because if the word v appears in one document the value is 1 and if it does not appear the value of v is 0. So, the difference is then 1. The weights of the words are then calculated / updated based on these differences. Initially these weights are set on zero. The weights of v are decreased if R and Hit have different values of word v, because it is not desirable that v separates two documents that belong to the same class. In case R and Miss have different values of word v, the weights are increased, because we want v to separate two documents that belong to different classes. The Relief algorithm is described in Figure 4.

One major shortcoming of Relief is that it does not eliminate redundant features, and thus therefore produces non-optimal feature subsets. Another shortcoming is that Relief is not able to deal with incomplete data and multi-class problems. This last limitation can be overcome by using ReliefF, an extension of Relief.

set all weights $W[\mathrm{v}] := 0.0$
for i := 1 to n do
  begin
    randomly select a document R_i*;*
    *find nearest hit (*Hit)
    *find nearest miss (*Miss)*;*
    for v := 1 to # words do

$$\mathrm{W[v]} := \mathrm{W[v]} - \frac{\mathrm{diff}(\mathrm{v}, \mathrm{R}_i, \mathrm{Hit})}{\mathrm{n}} + \frac{\mathrm{diff}(\mathrm{v}, \mathrm{R}_i, \mathrm{Miss})}{\mathrm{n}}$$

end

where

$$-\,\mathrm{diff}(\mathrm{v}, \mathrm{R}_i, \mathrm{Hit}) = \begin{cases} 0 & \mathrm{value}(\mathrm{v}, \mathrm{R}_i) = \mathrm{value}(\mathrm{v}, \mathrm{Hit}) \\ 1 & \textit{otherwise} \end{cases} \quad \textit{for nominal features}$$

- n *is a user - defined parameter*

Figure 4: The Relief algorithm as described in [5]

According **[5]** the space and time requirements for the Relief algorithm are $O(\#documents*\#words*n)$.

3.4 SVM

SVM stands for Support Vector Machine. A SVM is a hyperplane $w^T d + b$ that separates two classes. Parameters w^T and b are determined to maximize the margin. Documents on the boundary are called support vectors (see Figure 5).

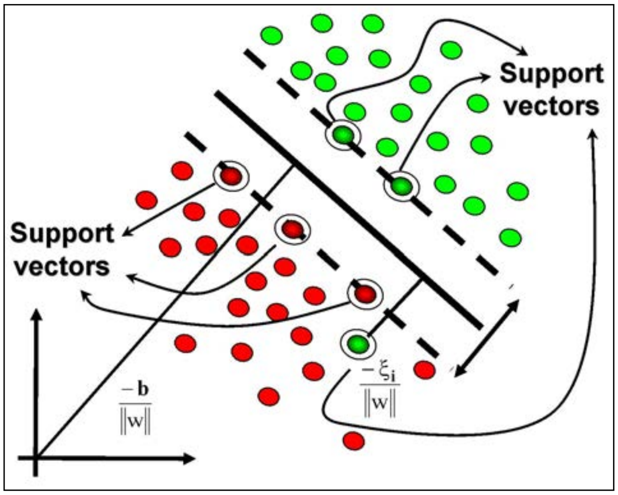

Figure 5: SVM[1]

SVM is mostly used for classification tasks and regression. It is a robust technique that shows superior performance and avoids overfitting **[28; 29]**. During this project SVM will not only be used as a classifier but also as feature selection technique. Using SVM as a feature selection technique is relatively new. In order to differentiate between the SVM classifier and the SVM feature selection technique, we will use the following notation: The notation SVM-Class is applied when the SVM as classifier is considered and SVM-FS when we mean the feature selection method. The SVM-FS algorithm is given in Figure 6. The basic idea behind this algorithm is that it first trains the SVM-Class using all the words. After that it obtains the weights of the documents. From these weights the weights of the words are calculated. The word(s) with the smallest weights are eliminated. After that it continues with training the SVM-Class using the remaining words. This process is repeated until all the words are eliminated. A low rank is assigned to words that are eliminated in the beginning, meaning that these words are of less importance. In the end a rank list is obtained with words.

[1] This picture is taken from http://www.cac.science.ru.nl/people/ustun/index.html

Initialize :

Subset of surviving words $\mathrm{s} = [1,2,\ldots,\mathrm{n}]$

Words ranked list $\mathrm{r} = [\ ]$

Repeat Step 1 - 7 until $\mathrm{s} = [\ ]$

1. *Restrict training examples to good words indices* $V = \mathrm{V}_0(:,\mathrm{s})$
2. *Train the classifier* $\alpha = SVM\text{ - }train\,(\mathrm{D}, C)$
3. *Compute the weight vector of dimension length*(s) *:* $\mathrm{W} = \sum_{\mathrm{d_i} \in \mathrm{D}} \alpha_\mathrm{i} \mathrm{c_i} \mathrm{d_i}$
4. *Compute the ranking criteria :* $rc(v) = (W(v))^2$, *for each word*
5. *Find the words with the smallest ranking criterion :* $f = argmin(\mathrm{rc})$
6. *Update words ranked list :* $\mathrm{r} = [\mathrm{r}, \mathrm{s}(f)]$
7. *Eliminate word with the smallest ranking criterion* $\mathrm{s} = \mathrm{s}[1 : f - 1, f + 1 : length(\mathrm{s})]$

Output :

Words ranked list r

Figure 6: SVM-FS algorithm as described in [8]

The $SVM\text{ - }train\,(D, C)$ algorithm finds the optimal hyperplane by solving the following quadratic optimization problem (see Figure 7). This algorithm calculates the weights α for the documents. Most of these weights are zero. The documents where these weights are non-zero are support vectors.

Minimize over α_k *:*

$$\mathrm{J} = \frac{1}{2} \sum_{\mathrm{hk}} \mathrm{c_h} \mathrm{c_k} \alpha_h \alpha_k (\mathrm{d_h}.\mathrm{d_k} + \lambda \delta_{\mathrm{hk}}) - \sum_{\mathrm{k}} \alpha_\mathrm{k}$$

subject to :

$$0 \le \alpha_k \le \mathrm{Q} \quad and \sum_k \alpha_k \mathrm{c_k} = 0$$

Output : α_k

Figure 7: SVM-train (D, C)

The soft margin parameters λ and Q are positive constants and $\delta_{hk} = \begin{cases} 1 & \text{if } h = k \\ 0 & \text{if } h \neq k \end{cases}$

These soft margin parameters allow a wider margin at the cost of misclassifying some of the documents.

According to the experiments done in **[8]** it takes 15 minutes to obtain the output when we have 2000 words and 62 documents and 3 hours when there are 7129 words and 72 documents. In most of the cases only a subset of the (training) data is taken to select words, because the training of the SVM model requires a lot of memory and CPU time **[2; 26]**. The standard complexity is about $O(D^{1.7})$ for SVM **[43]**.

3.5 Ripper

Ripper stands for Repeated Incremental Pruning to Produce Error Reduction. The Ripper algorithm first builds rules and then optimizes those rules. A rule is a condition and a condition is a conjunction of words. In the beginning of the building-phase it divides the (training) data into a growing set and a pruning set. The growing set is used to grow a rule / produce a rule, where the pruning set is used to prune the rule produced by the growing set. This rule is build based on the IG principle. If this rule satisfies certain conditions then this rule is added to the ruleset and the documents that are covered by this rule are deleted from the training set. This procedure is repeated until no positive documents are left over, or until the description length[1] (DL) of the ruleset and examples is 64 bits greater than the smallest DL met so far, or until the error rate >= 50%. After that the ruleset is optimized. For each rule in the ruleset the (training) data is divided into a new growing set and a pruning set. Two rules are then build, one new rule and one that adds other words to the existing rule. From these three rules (the one in the ruleset, the newly build rule, and the one that is an extension of the one in the ruleset) the final rule is chosen based on the minimum DL. Ripper uses a separate-and-conquer technique, because it finds a rule that can cover documents in the class, deletes those documents, and goes further with finding rules for documents that are left over. A detailed description of the Ripper algorithm is given in Figure 8.

[1] DL is the number of bits that are used to represent the model **[40]**

For each class C from smallest to largest
Build :
Split D into Growing and Pruning sets in the ratio 2 : 1
Repeat until there are no more uncovered documents of C, or the DL of ruleset and documents is 64 bits greater than the smallest DL found so far, or the error rate exceeds 50%
Grow phase : Grow a rule by greedily adding conditions until the rule is 100% accurate by testing every possible value of each word and selecting the condition with the highest information gain G
Prune phase : Prune conditions in last to first order. Continue as long as the worth W of the rule increases
Optimize :
Generate variants :
For each rule R for class C
Split D into a new Growing and Pruning set
Remove all documents from the Pruning set that are covered by other rules for C
Use the Grow and Prune phase to generate and prune two competing rules from the newly split data
R1 is a new rule, rebuilt from scratch
R2 is generated by greeding adding antecedents to R
Prune the rules using metric A on this reduced data
Select representative :
Replace R by whichever R, R1, R2 had the smallest DL

Figure 8: The Ripper algorithm as described in [40]

$$
\begin{aligned}
&G = p[log(p/t) - log(P/T)] \\
&W = (p+1)/(t+2) \\
&A = (p+n')/T \\
&p = \#\ \textit{positive documents covered by this rule} \\
&n = \#\ \textit{negative documents covered by this rule} \\
&t = p + n \\
&n' = N - n = \#\ \textit{negative documents not covered by this rule} \\
&P = \#\ \textit{positive documents of this class} \\
&N = \#\ \textit{negative documents of this class} \\
&T = P + N
\end{aligned}
$$

Figure 9: The meaning of symbols used in the Ripper algorithm as described in [40]

According to **[37]** the time complexity of Ripper is $O(\#\text{documents} * \log^2(\#\text{documents}))$.

3.6 BoosTexter

Boosting is a machine learning technique that performs categorization by joining simple and some inaccurate classifiers (e.g. rules) in order to find a highly accurate classification rule. Training of these rules is done sequentially; each rule is trained on those instances that were hard to categorize by the previous rules.

In **[1]** there are two extensions of the AdaBoost algorithm discussed: AdaBoost.MH and AdaBoost.MR. The goal of AdaBoost.MH is to predict only the correct classes, where the goal AdaBoost.MR is to rank the classes such that the highest rank is assigned to the correct classes. Only one of them will be used, namely the AdaBoost.MH with real valued predictions, because it outperforms all the other boosting algorithms (AdaBoost.MH with discrete predictions and AdaBoost.MR with discrete prediction). In case the size of the training set is smaller than thousand, the performance of Adaboost.MH is very poor. However, for large datasets the performance of Adaboost.MH is good. We will use this algorithm not for classification, but for feature selection.

In the first step of this algorithm the distribution of the weights of the documents is initialized. Then for each word the weights are calculated. This weight is calculated in a complex way. It considers 4 situations given a word v. One, the sum of the distribution of those positive documents is taken where the word is present. (U_+^1). Two, the sum of the distribution of those negative documents is taken where the word is present (U_-^1). Three, the sum of the distribution of those positive documents is taken where the word is absent (U_+^0). Four, the sum of the distribution of those negative documents is taken where the word is absent (U_-^0). After that the value of U_+^1 is multiplied by U_-^1 and the value of U_+^0 is multiplied by U_-^0. From both multiplications the sum Z_t is taken. This process is done for each word. After that the word with the smallest Z_t is then selected. It could be the case that there are more words that have the same smallest Z_t. In that case only one word is selected. This Z_t is among others used to update the distribution od the weights. After this has been updated the Z_t's are again calculated and a word with the smallest Z_t is then selected. This process repeats for several times. It depends on the user how many times it will be repeated. This process is described in details in Figure 10.

$$\textit{Initialize}: \mathrm{Dist}_1(i) = \frac{1}{(\#\mathrm{documents} * \#\mathrm{classes})}$$

$$\textit{For } t = 1,\ldots,T:$$

$$-\textit{Calculate } Z_t$$

$$-\textit{Choose word v with the smallest } Z_t \textit{ value}$$

$$-\textit{Update } \mathrm{Dist}_{t+1}(i) = \frac{\mathrm{Dist}_t(i)\exp\left(-C_i h_t(d_i)\right)}{Z_t}$$

where

$$h(d_i) = \begin{cases} q_0 & \text{if} \quad v \in d_i \\ q_1 & \text{if} \quad v \notin d_i \end{cases}$$

$$q_j = \frac{1}{2}\ln\left(\frac{U_+^j + \varepsilon}{U_-^j + \varepsilon}\right), j \in \{0,1\}$$

$$U_b^j = \sum_{i=1}^{\#\mathrm{documents}} \mathrm{Dist}_t\left[d_i \in \Gamma_j \wedge C_i = b\right], b \in \{+,-\} j \in \{0,1\}$$

$$\Gamma_0 = \{d : v \notin d\}$$

$$\Gamma_1 = \{d : v \in d\}$$

$$\varepsilon = \frac{1}{(\#\mathrm{documents}*\#\mathrm{classes})}$$

$$Z_t = 2\sum_{j\in\{0,1\}} \sqrt{U_+^j U_-^j}$$

$$T = \textit{a user}-\textit{defined value}$$

$$\mathrm{Dist} = \textit{distribution}$$

Figure 10: The AdaBoost.MH algorithm applied as feature selection method

It may happen that U_+^j or U_-^j is almost zero. In such cases q_j will become very large, which will lead to numerical problems. In order to avoid this, an ε has been added to both U_+^j and U_-^j.

According to **[1]** the space and time requirements per round t are $O(\#documents*\#classes)$ without including the calculation of U. The time required for the calculation of h is proportional to the total number of occurrences of all the words in the documents. Computing h can be very time consuming when the collection of documents is large **[1]**.

4 Implementation

For the implementation of the feature selection techniques different languages and software are used, namely Perl, C++ and Weka. Perl is a powerful language for text processing[1] that's why Perl is used for converting the text into a BOW model. Perl is also used to implement the odds ratio algorithm. This algorithm was already implemented in Perl by ParaBotS. However, as fast as Perl is in text processing, as slow it is in doing heavy (mathematical) computations. That is the reason why we could not limit ourselves to use Perl for the implementation of the other feature selection algorithms. C++[2] is a better language for doing heavy computations. The BoosTexter algorithm is one of those feature selection algorithms that requires a lot of computations, that is why we had decided to implement this algorithm in C++. Of course, one could argue why not use Matlab for implementing BoosTexter. The most important reason is that there is no specific interface between Perl and Matlab[3]. We have tried to work around, but it was not possible to call and gather Matlab from Perl in a smooth way. As Perl is used to access data it would require us to have an interface between Perl and the feature selection programs. Implementing the rest of the algorithms would require a lot of time. It was discovered that Weka already had an implementation of these algorithms. Weka is freely available Data Mining software written in Java[4] that contains machine learning algorithms that can be used for pre-processing, classification, regression, clustering, association rules, selecting attributes, and visualization. The feature selection techniques that will be used in Weka are InfoGainAttributeEval for the IG algorithm, ReliefFAttributeEval for the Relief algorithm, SVMAttributeEval for the SVM-FS algorithm, and JRip for the Ripper algorithm.

In Figure 11 a global scheme is provided for the implementation. Both steps are implemented in Perl. For storage, we used an MySQL database, accessed by the Perl SBI interface. Perl is thus not only used for pre-processing, but also as the main program. Step 1 in Figure 11 will be discussed in detail in Chapter 5 where the data is explained, because it belongs to the data conversion part. Step 2 in Figure 11 will be discussed in more detail in Chapter 7 Experimental set-up. In the end, when the results are obtained we will make use of the statistical freely available tool R. This tool will be used to analyze the results.

So, for this project the following languages and tools were used: Perl, C++, MySQL, Weka, and R.

[1] For more information see: http://perltraining.com.au/whyperl.html and http://www.perl.com/
[2] For more information see: http://www.cplusplus.com/
[3] http://www.mathworks.de/support/solutions/data/1-3UV21T.html?product=ML&solution=1-3UV21T
[4] Java is another programming language. For more information see: http://www.java.com/en/

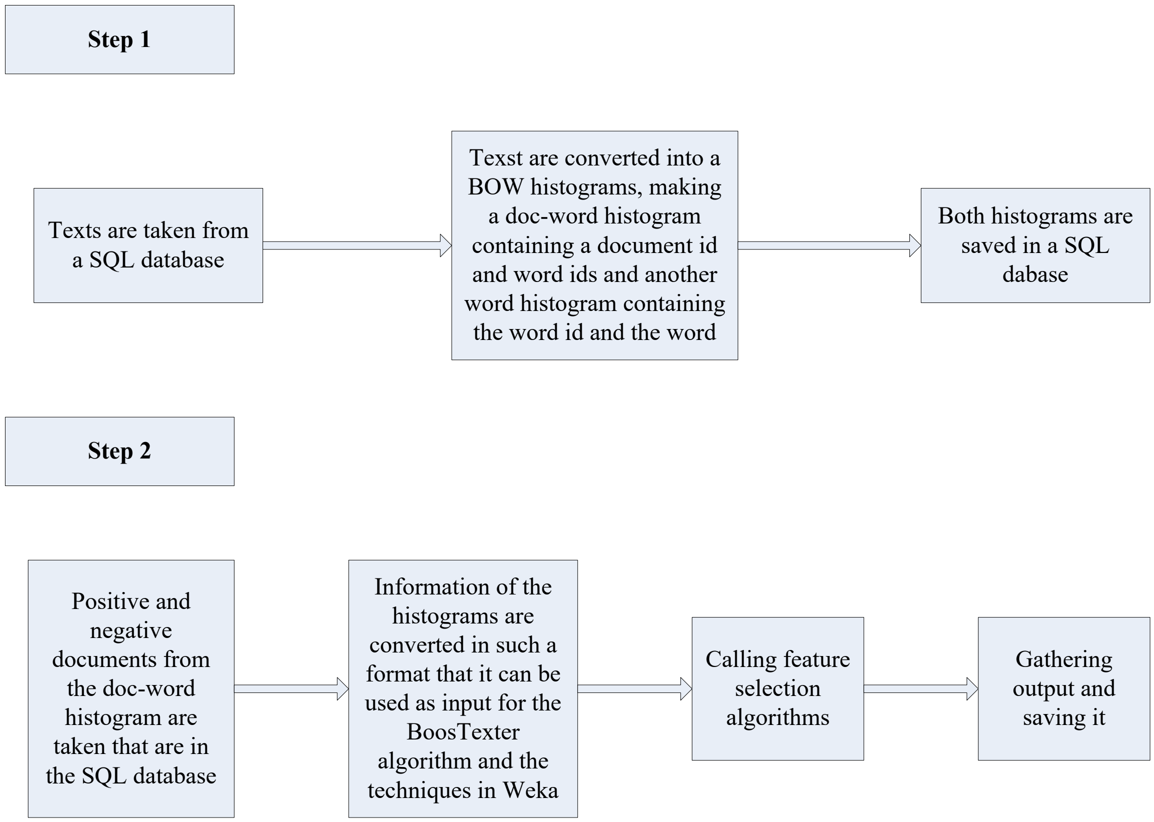

Figure 11: Global scheme of the implementation

5 Dataset

The data from the Vox-Pop database has been used. ParaBotS' Vox Populi application is able to figure out what is being said in a sentence by using natural language processing techniques, whether the message is positive or negative. By doing so for all the sentences on the thousands of pages it monitors everyday, the Vox-Pop is able to capture the general opinion. For a number of categories it shows the 5 entities (persons, or companies, or products) that were talked about either most positively or negatively that day. Plus the position they had yesterday. Currently, the vox-pop database contains five active categories: Celebrities (Celebs), Football (Voetbal), Political Parties (Partijen), Politicians (Politici), and Euronext. It would be impossible to consider all the entities in each category for this project, that is why a selection of category and entities are made. We decided to look at three of the five categories: Celebrities, Football, and Politicians. The category Political Parties basically covers the category Politicians in a general way, that is why it was decided not to use it. The other category, Euronext, was not used because at the moment there is a lot of information distributed via different media channels about banks due to the economic crisis world wide. So, we found it not interesting to look at this category. The entities were chosen based on Vox-Pop's half year review (historical data) that was available in the month July. For each selected category, except for the category Politicians, the first (top) two entities were taken and then two more or less random selected entities were chosen. For the category Politicians the entity Jan Peter Balkenende (which was on the top) was deselected, because it took a couple of weeks before we had any output[1]. This politician was replaced by another random politician. For the category Celebrity, the entities Britney Spears (BS), Madonna (M), Paris Hilton (PH), and Snoop Dogg (SD) were chosen, for the category Football the entities Edwin van der Sar (EvdS), Guus Hiddink (GH), Marco van Basten (MvB), and Wesley Sneijder (WS), and for the category Politicians the entities Ahmed Aboutaleb (AA), Ab Klink (AK), Geert Wilders (GW), and Rita Verdonk (RV). As the selected entities only exist of persons, this reference will also be used.

The information extracted from the internet is stored into tables in the Vox-Pop database. Not all the tables and the information in these tables will be discussed, but only the ones that were needed for this project. The first thing one needs to know is that each entity has an entity id that is stored in a table 'monitorentities'. Second, there is a table called 'newsitems' that contains a document id, the document's text, and the publication date. A Perl script is used to convert the texts from this table into two tables: a lexicon that contains words and word ids and a histogram table that contains document ids, word ids, and the frequencies of the word ids (see Step 1 in Figure 11). Texts, extracted from the table 'newsitems', are read line by line. In case a line contains more than 5 words, these words are first converted to lowercase words and then taken. We considered a line that contains less than 5 words not as a text that contains relevant information. This adjustment was necessary, because the data is currently unprocessed, i.e., the text

[1] When applying the SVM-FS to select words it took approximately 1 day before 1 cross-validation fold was finished. As our main goal was to find which technique produces better result, we decided to replace this entity for another one.

contains a lot of advertisement and tabs such as 'Radio & Video', 'Laatste reacties', 'Auto & Reizen', 'Meest gelezen artikelen', 'Stuur door', 'Reageer op dit artikel'. Of course there should be an advanced tool to process the data and moreover clean the data such that it only contains the relevant text. However, for the time being we choose to clean the data in a simple but not very effective way. The words that were taken from the text are then put into another table that contains word ids and the words. These words are only inserted in case they do not exist in the table. In case the word already exists in this table the id is taken. A word can appear more than once in a text, that is why we also keep up the frequency. For each text, the document id together with the word ids and the frequency are then inserted into a table called 'newsitemhistograms'. This process is repeated for each text in the category Politicians, Celebrities, and Football. So, there are two types of tables created for this project, one that contains documents ids, word ids, and word frequencies, and another that contains the word ids and the words itself. The table 'newsitems' contains a lot of documents. Not all these documents are so called 'relevant' for each entity. ParaBotS is measuring the relevance of a document, by assigning a score to these document ids. The scores for each document id is stored in the table called 'scores'. This table contains for each entity id and each document id the scores. Another table called 'x_monitoredEntities_categories' contains the entity id and the category id. So, for each category (id) only the document ids that have a score (for an entity) are selected and then processed further.

The Vox-Pop database contains approximately two years of data. This data consists of Dutch texts taken from several sites. Because of time constraint we decided to take only the data from the month June 2008. Taking data for more months would slow all experiments down. Our main research question will now slightly change by this decision. Instead of generating words that are characteristic for an entity, we are now generating words that are characteristic for an entity in a certain time period.

We will not only consider single words in a text, but also a combination of two consecutive words (bigrams). Sometimes looking at single words is not enough, that is why we will look at a combination of two consecutive words. For example, words like 'Los Angeles', 'Den Haag', 'Manchester United', 'Champions League', 'United States', 'New York', etc. only have meaning when they are taken together. Two consecutive words are joined by an underscore, e.g. 'New York' will be 'New_York' and because every word is first converted to lowercase the final word will look like 'new_york'. In a text there are many words that do not add anything when considered independently. Determiners are examples of such words. These words are also listed as stopwords. Examples of stopwords are, 'a', 'an', 'the', 'these', 'those'. When joining two consecutive words (making a doc-word histogram as in Figure 11) the stopwords are in a smart way taken out from the text. We had appoximately 500 stopwords, which would result in approximately 250,000 (500×500) stopwords. These stopwords would unnecessarily being added in the doc-word histogram. So, removing the stopwords before joining two consecutive words would save time and space. The statistics about the data from June 2008 are given in Table 2 (on category level) and Table 4 (on entity level) for single words and in Table 3 (on category level) and Table 5 (on entity level) for two consecutive words. Note that the length of the documents for single words includes stopwords. It should also be kept in mind that it may happen that for one or more persons there are no documents available. In this case nothing is written about that person. That is why when looking at the average documents per entity it may happen that this number is lower than we expected. The number of words in all documents divided by the number of documents is called 'Average length of the document'. When looking at Table 2 and Table 3 we see that the number of documents for each category is different. This difference can be explained by the type of words we are looking at. Suppose a document contains the following sentence "Her cat is big and old". From this sentence we have three single words (cat, big, and old) but no two consecutive words. If a document contains only this kind of sentences, then this document is meaningful for the single words data but not for the two consecutive words data. Therefore, it is possible that the number of documents for the two consecutive words may be less than the number of documents for the single words.

Category	# entities	# documents	Average documents per entity	Average length of the document
Celebrities	1903	2367	1	199
Football	902	4258	5	258
Politicians	177	9915	56	461

Table 2: Data from June 2008 for 3 categories for single words

Category	# entities	# documents	Average documents per entity	Average length of the document
Celebrities	1903	2331	1	38
Football	902	4242	5	48
Politicians	177	9877	56	80

Table 3: Data from June 2008 for 3 categories for two consecutive words

Entity	# documents	Average length of the document
Paris Hilton	42	106
Snoop Dogg	65	264
Britney Spears	92	138
Ahmed Aboutaleb	112	370
Madonna	128	188
Edwin van der Sar	247	360
Ab Klink	276	314
Wesley Sneijder	421	305
Guus Hiddink	506	289
Rita Verdonk	579	634
Marco van Basten	824	294
Geert Wilders	1119	524

Table 4: Data from June 2008 for 12 entities for single words

Entity	# documents	Average length of the document
Paris Hilton	42	21
Snoop Dogg	65	56
Britney Spears	91	33
Ahmed Aboutaleb	112	65
Madonna	128	46
Edwin van der Sar	247	66
Ab Klink	276	56
Wesley Sneijder	421	58
Guus Hiddink	506	55
Rita Verdonk	577	102
Marco van Basten	824	53
Geert Wilders	1118	107

Table 5: Data from June 2008 for 12 entities for two consecutive words

Note that all the documents where an entities' name appeared are called a positive universe / documents, i.e., these documents belong to the positive class. The rest of the documents where all other entities from the same category appeared are called a negative universe / documents, i.e., these documents belong to the negative class.

6 Evaluation Technique and Measures

6.1 Measuring distinctiveness and representativeness by classification

As discussed earlier we will use feature selection techniques in order to select few, representative, distinct, and relevant words. After these words are selected the question arises: How does one know where these words are representative, distinct, and relevant? We can measure the distinctive quality of our word lists by evaluating the performance of a machine learning algorithm that is based on only these words as features. Remember that before we selected these few words, we had positive and negative documents and also a large collection of words. From this (large) collection these words were selected. We can now use a machine learning technique (classifier / evaluation technique) to train these positive and negative documents with the selected words and then measure its performance. The question that arises is: How do we evaluate the performance of this classifier? This performance can be measured by calculating the accuracy, recall, precision, and F_1-measure. We will now discuss whether it is necessary to use all these four measures or just one of them. Suppose that we have a different ratio of positive and negative documents. For example, let us say 1 positive and 5 negative documents. If all documents are classified as negative we would have an accuracy of 80%. One could argue to use the same ratio, but then we would be faced with another problem, namely the feature selection technique is not able to select representative, distinct, and relevant words from a small negative universe and that using such a size would lead to a bigger chance of having words selected accidentally. So, it is out of the question to use accuracy as evaluation measure. Next are the precision and recall. These measures need to be considered together, because it could happen that many positive documents are classified as negative (where few negative documents as positive), which would result into a low recall and a high precision. However, if there are many negative documents classified as positive (and few positive documents as negative), then this would result in a low precision and a high recall. The F_1-measure is defined as a harmonic mean of the precision and recall. It is redundant to use the precision and recall if we can capture in one number both values. Therefore, we will use the F_1-measure as one of the evaluation measures. The formula for the F_1-measure is given in Figure 12.

$$F_1 - \text{measure} = \frac{2 * Precision * Recall}{Precision + Recal}$$

$$Precision = \frac{\# correctly\ classified\ documents\ of\ class\ A}{(\# correctly\ classified\ documents\ of\ class\ A + \# documents\ classified\ as\ belgonging\ to\ classA)}$$

$$Recall = \frac{\# correctly\ classified\ documents\ of\ class\ A}{\# documents\ in\ class\ A}$$

Figure 12: Formula F_1-measure

As discussed earlier we will need a machine learning technique to measure the distinctiveness and the representativeness of the words. This technique will evaluate each list of words that is produced by each feature selection algorithm. The evaluation technique that will be used is SVM-Class, the SVM as classifier. As mentioned earlier in Section 3.4 SVM-Class is a robust technique that shows high performance **[28; 29]**. The SVM-Class in Weka will be used.

Notice that the F_1-measure is just a number and the higher the number, the better the selected words can distinguish between the two classes. So, it is obvious that distinctive and even representative words are then selected. But, is the F_1-measure able to meet the relevance criterion? This is hard to say. We think that how relevant the selected words are, can be best judged by humans. This is why so called human scores come into the picture. How these human scores are calculated based on the selected words will be discussed into a separate Section.

6.2 Measuring representativeness and relevance by human judgment

In the previous section we discussed how we can measure the distinctiveness and even the representativeness of the selected words. We also observed that the relevance of the word lists could not be measured by a simple machine learning technique, but only by humans. The procedure of how the human will judge these word lists will be explained. Each person will get an entity word list containing single words and two consecutive words. They should then select for each entity word list 10 words that are most characteristic for that entity. After that the person should make a list of 5 words from the previous 10 words that are most characteristic for that entity. So, basically each person should make first a selection of 10 most characteristic words and of these 10 characteristic words also make a selection of only 5 most characteristic words.
Both lists (10 and 5 most characteristic words) should be handed in. But how does each person select the most relevant words for an entity? In order to select these words each person should open the site http://www.vox-pop.nl/, click on ‘Vox-Pop Halfjaaroverzicht’, and then for each entity read the headlines for the month June 2008, in this case the headlines of the weeks 23 until 27. Based on what is in the headlines and his or hers own knowledge, the person should select the 10 most characteristic words and of these 10 characteristic words select only 5 most characteristic words. From what is told above we can distinguish two steps: one how the entity word list is made from each technique and two how are the human scores calculated based on the selection of 10 and 5 relevant words.

How the entity word list is made will be illustrated with an example. Suppose we have 2 techniques, that each produced a list with 10 single words and 10 two consecutive words as the one shown in Table 6 and Table 7. From the single words and two consecutive words a list of 10 final words is made each ‘basically’ containing 5 single words and 5 two consecutive words. These final words are provided in Table 8. Then of these single words and two consecutive words for both techniques, one can make a list of distinct words i.e. the words that are selected by both techniques and thereby ignoring ‘repeated’ words. In this example, the distinct words are the one as shown in Table 9. This table of distinct words from the two techniques will be provided to each person for selecting 10 and 5 most characteristic words.

Technique 1	Technique 2
foto's	foto's
pussycat	federline
videoclip	lindsay
echt	album
emmy	amerikaanse
ex	zangeres
album	nieuw
los	emmy
amerikaanse	los
zusje	26-jarige

Table 6: Single words of two techniques of entity X

Technique 1	Technique 2
puppy_kopen	puppy_kopen
benji_madden	benji_madden
nicole_richie	nicole_richie
raar_trekje	raar_trekje
britney_spears	britney_spears
joel_madden	joel_madden
duitsland_spanje	duitsland_spanje
miljoen_euro	miljoen_euro
kate_beckinsale	kate_beckinsale
amerikaanse_tijdschrift	amerikaanse_tijdschrift

Table 7: Two consecutive words of two techniques of entity X

Technique 1	Technique 2
foto's	foto's
pussycat	federline
videoclip	lindsay
echt	album
emmy	amerikaanse
puppy_kopen	puppy_kopen
benji_madden	benji_madden
nicole_richie	nicole_richie
raar_trekje	raar_trekje
britney_spears	britney_spears

Table 8: Final list of words of two techniques of entity X

Distinct words
zusje
zangeres
videoclip
pussycat
nieuw
los
lindsay
foto's
federline
ex
emmy
echt
amerikaanse
album
26-jarige
benji
madden
amerikaanse_tijdschrift
benji_madden
britney_spears
duitsland_spanje
joel_madden
kate_beckinsale
miljoen_euro
nicole_richie
puppy_kopen
raar_trekje

Table 9: Distinct words of two techniques of entity X

After this list (see Table 9) has been provided to people, we will get the results back, each containing a list of 10 and 5 characteristic words. Suppose that we had only considered three persons for this experiment and that these persons select the following 10 and 5 characteristic words of this list as the one provided in Table 10. Based on these selected 10 and 5 characteristic words, the score for each distinct word is calculated as follows: if the word is not selected as one of the 10 most characteristic words it is assigned a 0, if the word is selected in the 10 most characteristic word list (and not as one of the 5 most characteristic words) it is assigned a 1, and if the word is selected as one of the 5 most characteristic it is assigned a 2 (see Table 11). Based on the total score for each distinct word, the Kendall's correlation coefficient (details see Appendix C) between this score and the words produced per technique is calculated (see Table 12). The Kendall's correlation coefficient for technique 1 for single words is -0.144, for two consecutive words is 0.177, and for composed words is 0.194.

<table>
<tr><td rowspan="2">Person 1</td><td>10 characteristic words</td><td>5 characteristic words</td></tr>
<tr><td>zangeres
foto's
federline
emmy
amerikaanse
26-jarige
benji_madden
joel_madden
nicole_richie
puppy_kopen</td><td>zangeres
amerikaanse
26-jarige
benji_madden
puppy_kopen</td></tr>
<tr><td rowspan="2">Person 2</td><td>10 characteristic words</td><td>5 characteristic words</td></tr>
<tr><td>zangeres
videoclip
linsay
amerikaanse
foto’s
benji_madden
joel_madden
nicole_richie
kate_beckinsale
puppy_kopen</td><td>zangeres
videoclip
nicole_ritchie
benji_madden
puppy_kopen</td></tr>
<tr><td rowspan="2">Person 3</td><td>10 characteristic words</td><td>5 characteristic words</td></tr>
<tr><td>zangeres
foto's
federline
album
lindsay
26-jarige
amerikaanse_tijdschrift
britney_spears
nicole_richie
puppy_kopen</td><td>zangeres
federline
26-jarige
benji_madden
puppy_kopen</td></tr>
</table>

Table 10: Selected words by 3 persons for entity X

Distinct words	Score Person 1	Score Person 2	Score Person 3	Total score of 3 Persons
zusje	0	0	0	0
zangeres	2	2	2	6
videoclip	0	2	0	2
pussycat	0	0	0	0
nieuw	0	0	0	0
los	0	0	0	0
lindsay	0	1	1	2
foto's	1	1	1	3
federline	1	0	2	3

ex	0	0	0	0
emmy	1	0	0	1
echt	0	0	0	0
amerikaanse	2	1	0	3
album	0	0	1	1
26-jarige	2	0	2	4
benji	0	0	0	0
madden	0	0	0	0
amerikaanse_tijdschrift	0	0	1	1
benji_madden	2	2	2	6
britney_spears	0	0	1	1
duitsland_spanje	0	0	0	0
joel_madden	1	1	0	2
kate_beckinsale	0	1	0	1
miljoen_euro	0	0	0	0
nicole_richie	1	2	0	3
puppy_kopen	2	2	2	6
raar_trekje	0	0	0	0

Table 11: Score of 3 persons on distinct words of two techniques of entity X

Distinct words	Average score of 3 Persons	Score Technique 1			Correlation coefficient of Technique 1		
		Single words	Two consecutive words	Composed words	Single words	Two consecutive words	Composed words
zusje	0	1	0	0	-0.144	0.177	0.194
zangeres	6	0	0	0			
videoclip	2	8	0	7			
pussycat	0	9	0	8			
nieuw	0	0	0	0			
los	0	3	0	0			
lindsay	2	0	0	0			
foto's	3	10	0	9			
federline	3	0	0	0			
ex	0	5	0	0			
emmy	1	6	0	4			
echt	0	7	0	6			
amerikaanse	3	2	0	0			
album	1	4	0	0			
26-jarige	4	0	0	0			
benji	0	0	0	0			
madden	0	0	0	0			
amerikaanse_tijdschrift	1	0	1	0			
benji_madden	6	0	9	3			
britney_spears	1	0	6	1			
duitsland_spanje	0	0	4	0			

joel_madden	2	0	5	0			
kate_beckinsale	1	0	2	0			
miljoen_euro	0	0	3	0			
nicole_richie	3	0	8	5			
puppy_kopen	6	0	10	10			
raar_trekje	0	0	7	2			

Table 12: Score of each (type of) word for technique 1 and the correlation

Note that the (average scores of the) 3 persons in this example is not a constant number. In reality more persons are approached.

6.3 Significance

For each technique one will have the F_1-measure and the score. We are considering 12 entities, meaning that we will have 12 F_1-measures and 12 total scores of humans. Note that an entity-technique will from now on mean an entity and within an entity a technique. If we want to select the technique that is best in representing representative and distinct words, we could simply look at the highest F_1-measure. But what if this measure does not differ that much between techniques? In this case we would need a statistical measure to determine if there is a significant difference between these techniques. This will be done by using ANOVA (Analysis of Variance) (details see Appendix D). If there are no significant difference between the techniques, one can simply select the best technique, by not only looking at the highest F_1-measure, but also taking into consideration the time required to obtain these words. The null hypothesis when using ANOVA is that there is no difference between the techniques. This null hypothesis will be rejected for a p-value smaller than 0.05.

For scores assigned by humans, the Kendall's correlation coefficient will be calculated between the total score for each word and the words produced by each technique see Table 12). Next to this, the Kendall's test will be applied to check wheter this coefficient is significant, i.e., if the null hypothesis stating that there is no correlation is rejected. The null hypothesis is rejected for p-values smaller than 0.05. If we apply this to our example in Table 12, we obtain the following p-values 0.391, 0.290, 0.250 for respectively single, two consecutive, and composed words. Based on these p-values we cannot reject the null hypothesis. So, we cannot assume that there is a correlation between the words selected by humans and the ones selected by the technique.

6.4 Measure stability by nominal concordance

Besides evaluating the words, it is also interesting to see how stable each feature selection algorithm is given different negative documents. This stability can be measured by calculating the nominal concordance. The nominal concordance is a measure proposed by Marten den Uyl.

Suppose we have positive documents and n different samples of negative documents. Each sample can be used as negative universe for the positive documents. A feature selection technique can then select few, representative, distinct, and relevant words. If we use all the n samples, then we get n times a selection of words. Let us assume for the moment that n is equal to 2 and also that each time the same number of words is selected, meaning that both list contain exactly the same number of selected words. We now have 2 lists of selected words; each generated using a different set of negative documents. If the feature selection technique is very stable, these two lists should not differ too much. In the best case these 2 lists would be exactly the same (resulting in a nominal concordance of 1). The nominal concordance thus measures the number of words that are the same in both lists normalized by the total number of words that could be the same. Another example, suppose n is now equal to 3. And that using the first sample of negative documents 10 words are selected, using the second sample 10 words, and using the third sample 7 words. There are 3 combinations possible: one compare the first list with the second list of words, two compare the first list with the third list of words, and three compare the second with the third list of words. The total number of words that could be the same is equal to $10+7+7=24$. Because the third list of words only contains 7 words, there are only 7 words that could be maximal the same when this list is compared to other list. Let us assume now that when comparing the first list with the second list there are 8 words that are the same, when comparing the first list with the third list there are 7 words the same, and when comparing the second list with the third list there are 5 words the same. The nominal concordance is thus equal to $(8+7+5)/24=0.83$. This can be summarized into the following formula:

$$\boxed{\textit{Nominal concordance} = \frac{\#\textit{concordances}}{\#\textit{max concordances possible}} = \frac{\#\textit{concordances}}{\sum_{i=1}^{n}\sum_{j>i}^{n} \min\ \#\textit{words}(i,j)}}$$

Figure 13: Formula for the nominal concordance

The concordance is the number of words that is the same. In case each time the same number of words is selected, this formula can be simplified to:

$$\boxed{\textit{Nominal concordance} = \frac{\#\textit{concordances}}{0.5 * n(n-1) * k}}$$

Figure 14: The nominal concordance in case the number of selected words is always the same

7 Experimental set-up

The Experimental set-up is discussed, as the chapter's name already indicates. As explained in Chapter 4 texts are converted into BOW histograms where each document contains the word ids of the words that were in the text and their frequencies. We used word ids instead of words, because these ids are integers and can be compared more easily than strings. The documents with their word ids are then put into a SQL database in a histogram format. This is done for each category (Celebrities, Football, and Politicians). Then a Perl script is called which for each entity selects the documents that belong to that entity (positive documents) and the rest of the documents (negative documents). We decided to use stratified 5-fold cross-validation and a maximal of two negative random samples where each sample is 5 times bigger than the size of the positive documents. It could be the case that the size of the negative documents is many times bigger than the size of the positive documents. Using all these negative documents would then negatively influence the selection of words, that is why we decided to use samples of the negative documents. Meaning, we take a sample of the negative documents. The number of samples we will take is maximal 2. The choice of two random samples was because the results of using one sample could be based on coincidence. We did not consider more than two samples because of time constraints. In order to select proper words that are characteristic for the entity, one also needs to have a suitable size of the negative universe, that is why it was chosen that the size of the negative documents should be 5 times bigger than the size of the positive documents. Thus, each sample taken from the negative universe should contain 5 times more documents than the positive universe. However, in order to reduce any bias caused by choosing a particular sample set, we need to do a stratified 5-fold cross validation. Stratified sampling means that the documents of each class are in the same ratio present in the training and test set.

As one can imagine, there are many words present in all the selected documents. It can vary from hundreds to thousands. It is intuitively clear that not all words are informative, that is why we decided to eliminate from all the words the stopwords and the entity's name. After that with odds ratio a selection of 200 words was made, each word containing a high odds ratio of belonging to the positive universe. The size of the random samples was then decided, i.e., is it 1 or 2? (As explained earlier this could be maximal 2.) Since we are not using all the negative documents, it could happen that each time we get only the documents from the first days or weeks. That is why we decided that the selection of the negative documents should be done randomly. Meaning, that all the negative documents should first be randomized before taken a sample of it. In MySQL this can be easily done with the command "order by rand()".

So, now we have the 200 words with the highest odds ratio for the positive universe. The selection of these 200 words is done for single and two consecutive words. Then for each random sample and each cross validation training and test set are made, where the training set is provided to the feature selection techniques (BoosTexter, IG, Ripper, Oddsratio, Relief, and SVM). Using this training set, the top 10 words are selected by each technique. Now, the training and test set are changed such that they only contain these 10 words instead of the 200 words. Meaning, the rows (documents) stay the same where the columns of the 190 features are removed. These training and test set are then provided to the SVM-Class to calculate the F_1-measure. The training and test set are transformed into arff files within the Perl script. Since words that are higher in the list are more important, we decided to assigned weights to them. The first word will get a weight of 10, the second word a weight of 9, the third word a weight of 8, …, and the last word a weight of 1. This weight will only play a role when selecting the 10 final words for each random sample and eventually for the selection of final 10 words for an entity. Everything that is described so far can be viewed in Figure 15. The code for calling BoosTexter, the feature selection techniques in Weka, and the SVM-class in Weka is provided in Appendix E. We used 100 iterations in BoosTexter to select the 10 words with the highest weight within each cross validation fold.

Now, we have calculated the F_1-measure for each cross validation for each single word and two consecutive words, we need to calculate the F_1-measure for joined single and two consecutive words. First we explain how 10 single words and 10 of two consecutive words are merged such that we obtain 10 words that contains both single and two consecutive words. For each single word in the top 10 it is looked up whether there exists a combination of this word in the two consecutive words. If there exist a combination, than regardless of its position in the top 10, the two consecutive words is taken. After that it is checked how many words are needed to obtain 10 words containing both single and two consecutive words. This number is equal to 10 minus the words that are already selected. For the remaining single words and two consecutive words and equal size of words is taken such that in the end 10 final words are obtained that consists of single and two consecutive words. These 10 final words obtained will from now on called, 10 composed words. So, we have 10 single words, 10 two consecutive words, and 10 composed words. From this last list of words we need to compute the F_1-measure. For each cross validation fold we have the documents that are in the training and test set for the single and two consecutive words. Basically, documents from the two consecutive words are a subset of the documents from the single words[1]. For this reason we could satisfy by only looking at the documents from the single words. For each word in the composed word it is then it checks in which of the documents it exists. In such a way the training and test set are created for the 10 composed words for each cross validation. Note that the training documents in each fold for the single words and the two consecutive words are the same, except that some of the documents of the single words may not exist in the two consecutive doc-word histogram. Given the training and test set for the 10 composed words, we are now able to calculate the F_1-measure (on the test set) using SVM-Class.

For the selection of the 10 final words for an entity we basically take the highest sum of the words within the 5-fold cross validation (obtaining the 10 final words for a random sample), and then if there are more than one random samples, take the highest sum of the words within the random samples. As you all may know by now the 10 words are

[1] Imagine that a document contains all stopwords with one single word then this document is represented in the single word doc-word histogram, but not in the two consecutive doc-word histogram.

produced by a technique, so we basically obtain for each entity-technique the 10 final words. This procedure is done for single, two consecutive, and composed words. The list of these words can be found in Appendix A. As discussed in Section 6.2 a list will be made for each entity, containing distinct words. This list will then be provided to humans. There were 18 (out of the 30) persons who where able to return 10 and 5 characteristic words for each entity. Based on these results, a score can then be assigned to each technique. The total score for each word will then be calculated. Based on the words selected by humans and the ones selected by the techniques the Kendall's correlation coefficient will be calculated. The Kendall's correlation coefficient will check whether there is a positive correlation between what humans think and what the techniques produce. We will also look whether this coefficient is significant by applying the Kendall's test. For p-values smaller than 0.05 the null hypothesis is rejected, i.e., we can assume that there is a significant correlation of words selected by humans and the ones selected by techniques. For each technique and for each type of word (single, two consecutive, and composed) it will be computed how many times the Kendall's correlation coefficient was significant. The technique that has the highest number will then be advised. As we are considering 12 entities this number can maximal be 12. Next to this, we are also interested in the Kendall's correlation coefficient between words selected by humans, i.e., do the persons agree with eachother, is each person selecting the same words or not? Suppose that we have only 3 persons than we can calculate the correlation coefficient between person 1 and 2, between person 2 and 3, and between person 1 and 3. For each person pair the Kendall's correlation coefficient will be calculated and in the end the average will be taken. Next to this, we will measure the significance of the correlation coefficient obtained by each person pair. The ratio, the total number of significant correlation found divided by the maximum number of siginificant correlation coefficient possible, will be provided in the next chapter. The correlation coefficient found between humans mutally will then be compared with the correlation coefficient found between humans and techniques. Techniques are performing better when the correlation coefficient between humans and techniques is larger than the one found between humans mutually.

As discussed earlier in Section 6.3 we will apply ANOVA to test whether there is a significant difference between these techniques, given the F_1-measure. If there are no significant differences between the techniques, one can simply select the best technique, by not only looking at the highest F_1-measure but also taking into consideration the time required to obtain these words. A global estimate of the computational time for each entity-technique for single and two consecutive words are given in the next chapter. A more detailed result concerning the computational time can be found in Appendix B.

Next to producing few, representative, distinct, and relevant words, one also would like to find out how stable each feature selection technique is. This stability can be measured by nominal concordance. Nominal concordance will be measured between words obtained by various samples. In order to have a meaningful number of it, we need to have at least 5 random samples. However, not for every entity there are (at least) 5 random samples available, as one can calculate for itself. Only the entities Paris Hilton, Snoop Dogg, Britney Spears, Ahmed Aboutaleb, and Ab Klink have at least 5 random samples.

Therefore, we will only measure the nominal concordance for these 5 entities. The results are provided in the next Chapter. Note that the procedure for obtaining the final words for each random sample is not changed.

We will use the following notation from now on:

- Single words – SW
- Two consecutive words – TCW
- Composed words – CW

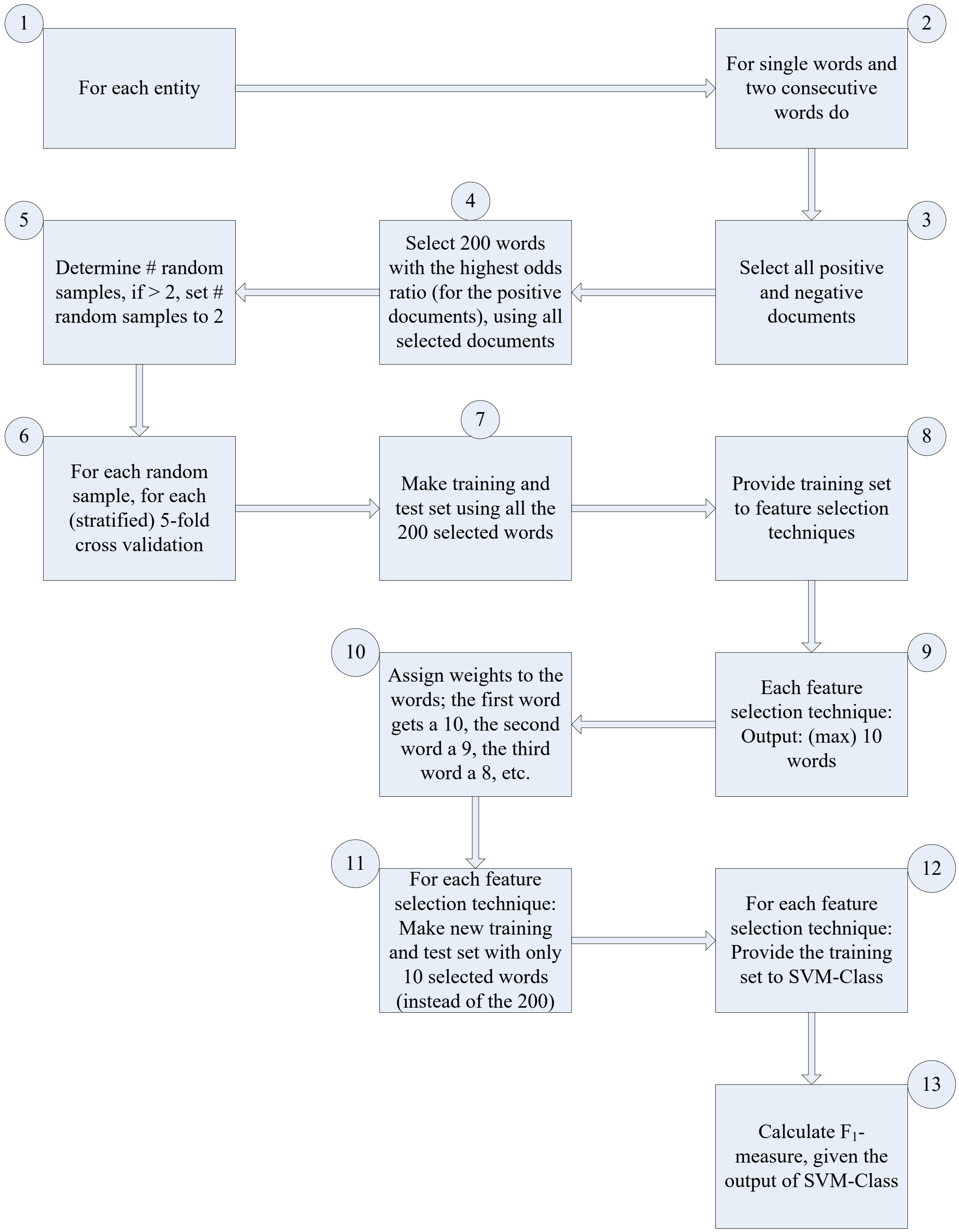

Figure 15: Steps for the selection of 10 words and the calculation of the F_1-measure

8 Results

8.1 F_1-measure of SVM

For each entity-technique the average F_1-measure is provided with its corresponding standard deviation. This is done for single, two consecutive and composed words. The entities are ordered by the number of documents in Table 13.

Entity	Technique	Average F_1-measure			Standard deviation F_1-measure		
		SW	TCW	CW	SW	TCW	CW
PH	BoosTexter	0.63	0.66	0.53	0.15	0.15	0.15
	IG	0.51	0.50	0.48	0.23	0.18	0.25
	Oddsratio	0.43	0.47	0.44	0.13	0.17	0.14
	Relief	0.27	0.41	0.40	0.22	0.18	0.26
	Ripper	0.59	0.55	0.61	0.12	0.08	0.14
	SVM	0.66	0.52	0.58	0.22	0.16	0.19
SD	BoosTexter	0.10	0.17	0.15	0.11	0.08	0.12
	IG	0.17	0.12	0.21	0.10	0.12	0.09
	Oddsratio	0.01	0.09	0.12	0.04	0.12	0.11
	Relief	0.09	0.03	0.07	0.09	0.08	0.09
	Ripper	0.19	0.12	0.16	0.11	0.10	0.08
	SVM	0.33	0.13	0.29	0.11	0.10	0.09
BS	BoosTexter	0.79	0.83	0.82	0.05	0.04	0.04
	IG	0.78	0.82	0.79	0.06	0.05	0.08
	Oddsratio	0.68	0.78	0.79	0.11	0.08	0.08
	Relief	0.21	0.75	0.61	0.16	0.05	0.11
	Ripper	0.79	0.83	0.83	0.05	0.03	0.03
	SVM	0.84	0.84	0.83	0.06	0.05	0.04
AA	BoosTexter	0.87	0.94	0.93	0.03	0.04	0.04
	IG	0.89	0.87	0.88	0.03	0.03	0.04
	Oddsratio	0.88	0.87	0.87	0.03	0.03	0.03
	Relief	0.87	0.88	0.87	0.02	0.03	0.03
	Ripper	0.89	0.94	0.93	0.03	0.04	0.03
	SVM	0.89	0.93	0.92	0.03	0.04	0.03
M	BoosTexter	0.11	0.45	0.24	0.07	0.08	0.11
	IG	0.42	0.40	0.37	0.10	0.10	0.08
	Oddsratio	0.20	0.38	0.36	0.12	0.08	0.08
	Relief	0.19	0.26	0.20	0.12	0.09	0.09
	Ripper	0.42	0.44	0.43	0.11	0.07	0.06
	SVM	0.54	0.46	0.51	0.11	0.11	0.09
EvdS	BoosTexter	0.25	0.42	0.37	0.13	0.08	0.10
	IG	0.52	0.46	0.51	0.09	0.07	0.05

	Oddsratio	0.51	0.39	0.51	0.05	0.09	0.05
	Relief	0.11	0.07	0.11	0.11	0.08	0.11
	Ripper	0.43	0.45	0.50	0.07	0.09	0.07
	SVM	0.50	0.40	0.53	0.04	0.05	0.04
AK	BoosTexter	0.95	0.62	0.94	0.01	0.08	0.03
	IG	0.95	0.56	0.95	0.01	0.13	0.01
	Oddsratio	0.95	0.59	0.46	0.01	0.09	0.12
	Relief	0.95	0.23	0.95	0.01	0.13	0.01
	Ripper	0.95	0.74	0.95	0.01	0.08	0.01
	SVM	0.95	0.67	0.92	0.02	0.14	0.10
WS	BoosTexter	0.25	0.45	0.41	0.09	0.05	0.08
	IG	0.39	0.46	0.44	0.10	0.06	0.05
	Oddsratio	0.07	0.45	0.29	0.15	0.07	0.07
	Relief	0.00	0.14	0.03	0.00	0.15	0.10
	Ripper	0.29	0.47	0.46	0.10	0.07	0.06
	SVM	0.43	0.31	0.44	0.05	0.06	0.05
GH	BoosTexter	0.63	0.49	0.48	0.09	0.04	0.18
	IG	0.75	0.54	0.74	0.04	0.05	0.01
	Oddsratio	0.73	0.46	0.61	0.01	0.03	0.27
	Relief	0.75	0.05	0.72	0.05	0.05	0.02
	Ripper	0.74	0.53	0.71	0.05	0.06	0.04
	SVM	0.76	0.41	0.70	0.04	0.06	0.05
RV	BoosTexter	0.64	0.52	0.61	0.10	0.04	0.07
	IG	0.74	0.49	0.67	0.02	0.02	0.06
	Oddsratio	0.67	0.47	0.60	0.04	0.01	0.04
	Relief	0.59	0.00	0.13	0.04	0.00	0.22
	Ripper	0.72	0.55	0.73	0.02	0.03	0.03
	SVM	0.76	0.50	0.72	0.02	0.05	0.08
MvB	BoosTexter	0.60	0.42	0.59	0.03	0.13	0.03
	IG	0.60	0.37	0.54	0.03	0.05	0.04
	Oddsratio	0.61	0.54	0.60	0.03	0.03	0.03
	Relief	0.44	0.00	0.61	0.25	0.00	0.05
	Ripper	0.61	0.45	0.56	0.03	0.12	0.09
	SVM	0.58	0.34	0.55	0.06	0.03	0.11
GW	BoosTexter	0.70	0.71	0.71	0.06	0.03	0.03
	IG	0.66	0.40	0.70	0.12	0.03	0.03
	Oddsratio	0.70	0.40	0.65	0.03	0.03	0.07
	Relief	0.64	0.40	0.63	0.05	0.03	0.05
	Ripper	0.79	0.71	0.79	0.03	0.03	0.02
	SVM	0.81	0.69	0.78	0.02	0.09	0.02

Table 13: F_1-measure for 12 entities

In order to visualize the results in Table 13 box-plots are made for single, two consecutive and composed words (see Figure 16). From the box-plots for single words we can see that the F_1-measure for each technique does not differ much. This observation is also confirmed when applying ANOVA. We get a p-value of 0.30, which means that we cannot reject the null-hypothesis that states that there is no difference between the techniques. This implies that there is no significant difference between the 6 feature selection techniques. On the other hand, when looking at the box-plots for two consecutive words, we see that these differ per technique. If we apply ANOVA on this data we get a p-value of 0.02. This means we can reject the null-hypothesis. So, our observation is confirmed. Taking a closer look to these box-plots, it seems that the Relief algorithm is the one that causes this difference. If we ignore / take out the F_1-measures for this algorithm and apply ANOVA on the rest of the 5 algorithms, we get a p-value of 0.87. This indicates that there is no significant difference between these 5 algorithms (BoosTexter, IG, Oddsratio, Ripper, and SVM) if we look at the F_1-measure. The last box-plots are of the composed words. We can observe that these box-plots do not differ much per technique, which is also confirmed with ANOVA that gives a p-value of 0.31.

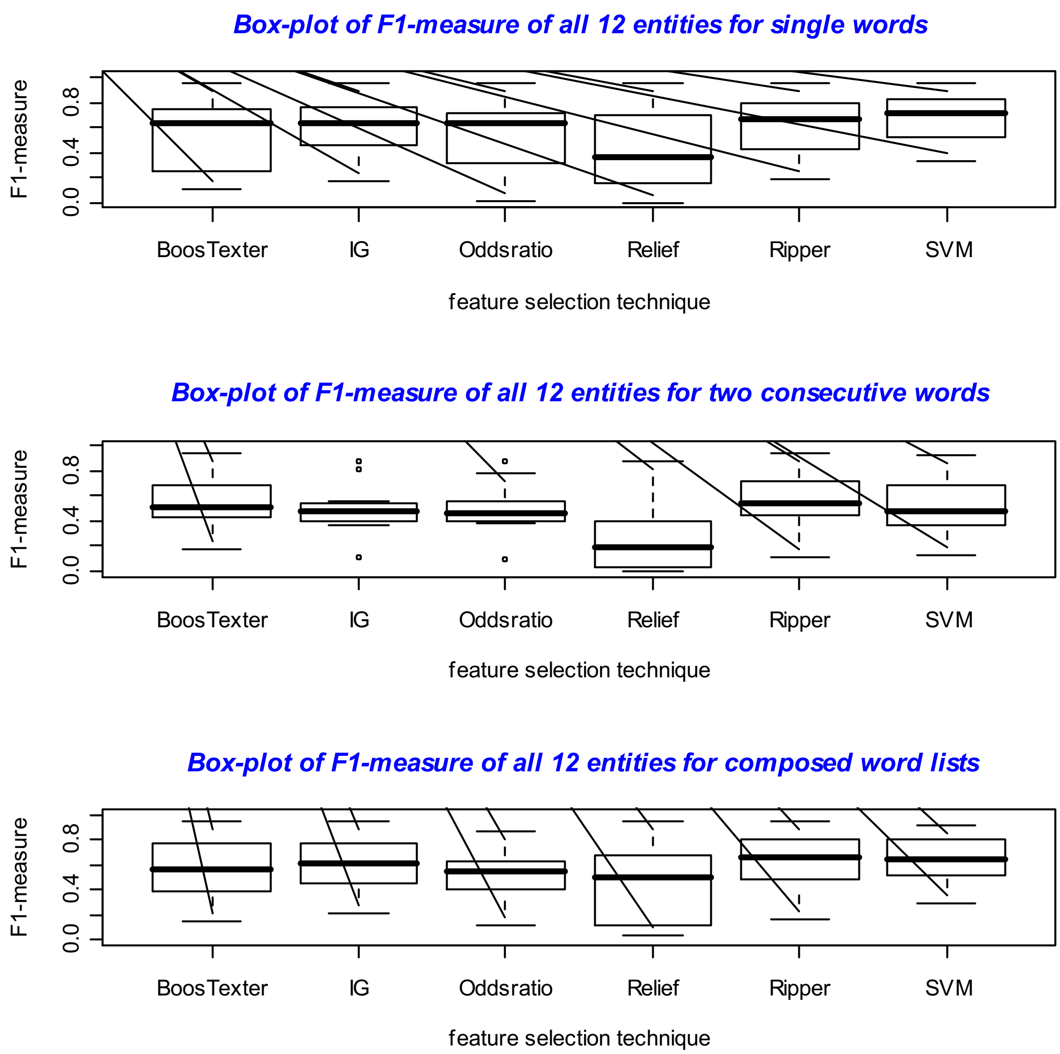

Figure 16: Box-plots of F_1-measure for SW, TCW, and CW

8.2 Correlation between techniques and humans

First we are interested in the average correlation (coefficient) of words selected by each person, i.e., we are interested in the correlation coefficients between humans mutually. For each person pair this correlation coefficient is computed and the average is taken. Also, when this correlation coefficient is computed it is checked whether it is significant or not. As there are 18 persons, the number of significance can be maximal 153. The ratio, the total number of significant correlation found divided by the maximal correlations possible is provided in Table 14 together with the average correlation coefficient.

Entity	Kendall's correlation	
	Average correlation coefficient	Significance Ratio
PH	0.35	0.73
SD	0.46	0.93
BS	0.32	0.56
AA	0.25	0.54
M	0.35	0.80
EvdS	0.45	0.87
AK	0.27	0.56
WS	0.37	0.82
GH	0.25	0.50
RV	0.19	0.39
MvB	0.31	0.65
GW	0.39	0.77

Table 14: Average Kendall's correlation coefficient for 18 persons

The data from Table 14 more or less suggests that there is a linear relationship between the significance ratio and the average correlation coefficient. Therefore, these values are plotted against eachother (see Figure 17). This is of course obvious, the larger the coefficient is, the more likely it is that this coeffient is significant, .i.e., that we can reject the null hypothesis stating that there is no correlation. In case there is a complete agreement between the 18 persons the correlation coefficient will be 1. If there is a complete disagreement between the 18 persons then the correlation coefficient will be -1. In case the persons randomly (dis) agree, the correlation coefficient is 0. It seems that most persons have a different opinion about Rita Verdonk (average correlation coefficient of 0.19), and that most persons strongly agree on what is typical for Snoop Dogg (average correlation coefficient of 0.46) which is extremely surprising, because almost everybody complained about the fact that they did not know which words to select for Snoop Dogg. So, it was more likely that each person would select random words. It seems however that the less choice a person has for selecting characteristic words the better they agree on which to select. Test persons also strongly agree (> 80 %) on which words were characteristic for Madonna, Edwin van der Sar, and Wesley Sneijder. The entities where people less agree on are Britney Spears, Ahmed Aboutaleb, Ab Klink, and Guus Hiddink. The sifnificant ratio varies is here around the 0.50, which means that only half of the correlations where significant. The rest of the entities Paris Hilton, Marco van

Basten, and Geert Wilders have a significance ratio (and correlation) that suggest a slight agreement between persons (0.65 -0.77).

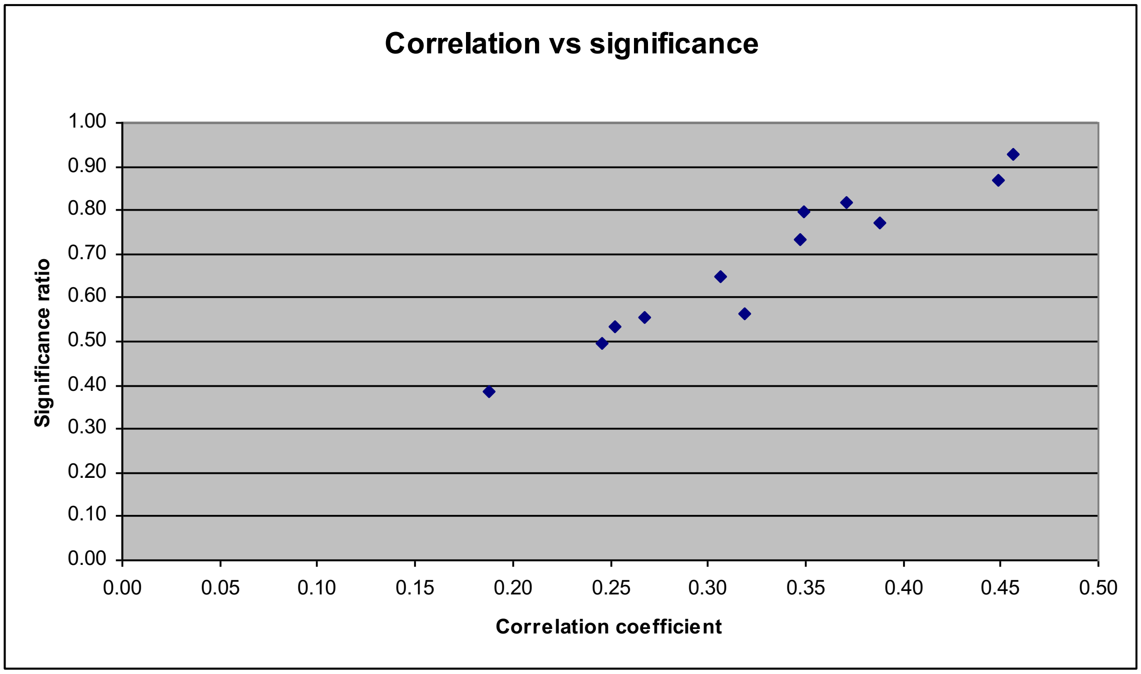

Figure 17: Kendall's correlation coefficient vs the significance ratio

In Table 15 two word lists are provided, namely one for the entity Rita Verdonk and Snoop Dogg. These two entities are chosen, because they correspond to the best and the worst agreement among test persons.

List provided to humans	
Snoop Dogg	*Rita Verdonk*
wenen_duitsland	zware_persoonsbeveiliging
welkom	zetels_halen
waaronder	wouter_bos
vrij	woordvoerder_kay
verenigde_staten	wilders
vat	werk
vari_rend	vvd-fractievoorzitter_mark
tweede_kamer	voorzorg_binnen
tweede	verenigde_staten
tu_delft	tweede_kamerlid
tori_spelling	tweede_kamer
tomtom	tweede
thomas_berge	tv-programma_knevel
sylvie_viel	trots
rijden	tournee
rechtbank	ton

rapper	tienduizenden_euro's
overwinning	terrorismebestrijding_nctb
overbelast_raakt	stapt_volgende
opnemen	sinke
ontvangen	rdinator_terrorismebestrijding
olieprijs	rdinator
olie	probleem
nieuw_middagnieuwsbrief	politieke_partijen
nieuw_album	politieke_beweging
nicolas_sarkozy	politieke
new_york	politica
nederland	persoonsbeveiliging
music_hall	persoonlijk_adviseur
music	peiling
missy_elliott	partij
miljoen_euro	onderzoeker_maurice
marks_brengt	nooit
maak_acteurs	nina_brink
londense_luchthaven	nederland_ton
leuke	nederland
jongeren	nctb
jongen	nationaal_co
john_marks	minister_ernst
jan_smit	minister
iran	miljoen_euro
iraanse_bank	maxime_verhagen
iemand	man
hoog_niveau	mail_artikel
hogere	kamer
heineken_music	kabinet
heineken	jan_marijnissen
grootste_iraanse	inmiddels
goed	hirsi_ali
gisteren	hirsch_ballin
gerard_joling	hand
georgina_verbaan	haag
ge_nteresseerd	groenlinks
frans_bauer	goed
europese_ministers	gehouden_vanwege
europese	geert_wilders
euro	ge_nformeerd
ek_jan	ernst_hirsch
ek	dreiging
eiland_aruba	den_haag
druk_momenteel	den_brink

druk	den
dertig	co_rdinator
daalde	buitenlandse_zaken
com	brink
binnenkort	binnen
behalve	beweging_trots
ballistische_raketten	beweging
amy_winehouse	beveiliging
amsterdam	adviseur
amerikaanse_rapper	
amerikaanse_ministerie	
amerikaanse	
altijd	
allemaal	
acteurs_zetten	
aandeel_noteerde	

Table 15: Distinct word list for the entities SD and RV

If we would choose from Table 15 the most characteristic words, then these words would probably be ones that are highlighted. There were 78 words available for Snoop Dogg from where we could select the 10 characteristic words. From these 78 words, only 30 words were selected overall by all the test persons (38%). From these 30 words only 7 words were selected only once and 3 words where selected twice. The 20 words that are selected more than 3 times are highlighted in Table 15. For Rita Verdonk there were 70 words available. From these 70 words, 40 words were selected overall by all the test persons (57%). There where 28 words that were selected more than 3 times. So, it seems that it is easier to select the words for Snoop Dogg than for Rita Verdonk, because for Snoop Dogg there the list contained more rubbish than for Rita Verdonk.

We did not only use the average F_1-measure as evaluation measure, but also the correlation between the words selected by humans and the words produced by each feature selection technique. This correlation coefficient is provided in Table 16 together with the p-values. These p-values state whether the correlation coefficient is significant, i.e., can we reject the null hypothesis that states that there is no correlation? The null hypothesis is rejected for p-values smaller than 0.05. So, in case a p-value is smaller than 0.05 we can assume that there is a correlation between words selected by the technique and the humans. Note that the total score for each word is provided in Appendix G.

Entity	Technique	Kendall's correlation			P-values		
		SW	TCW	CW	SW	TCW	CW
PH	BoosTexter	0.13	0.21	0.14	0.367	0.211	0.215
	IG	0.08	0.09	0.01	0.576	0.594	0.913
	Oddsratio	-0.12	-0.07	0.03	0.386	0.706	0.781
	Relief	0.47	0.31	0.24	0.001	0.062	0.031
	Ripper	0.06	0.29	0.07	0.705	0.079	0.513
	SVM	-0.01	0.11	0.01	0.957	0.506	0.929

SD	BoosTexter	0.24	-0.04	0.16	0.086	0.767	0.119
	IG	-0.03	0.37	0.03	0.830	0.009	0.740
	Oddsratio	0.11	0.05	0.12	0.463	0.725	0.254
	Relief	0.06	0.10	0.22	0.707	0.505	0.034
	Ripper	0.29	0.07	0.07	0.043	0.630	0.504
	SVM	0.00	-0.20	-0.17	1.000	0.171	0.092
BS	BoosTexter	-0.18	0.36	0.37	0.272	0.030	0.002
	IG	0.03	0.46	0.26	0.896	0.006	0.029
	Oddsratio	-0.17	0.31	0.14	0.321	0.063	0.260
	Relief	-0.01	0.47	0.23	0.979	0.004	0.051
	Ripper	-0.02	0.27	0.35	0.917	0.108	0.003
	SVM	-0.17	0.01	0.08	0.321	0.955	0.493
AA	BoosTexter	0.10	0.14	0.24	0.504	0.389	0.027
	IG	0.25	0.25	0.19	0.089	0.107	0.080
	Oddsratio	-0.09	0.21	0.16	0.541	0.188	0.138
	Relief	0.13	0.23	0.18	0.391	0.147	0.104
	Ripper	0.37	0.19	0.17	0.010	0.249	0.113
	SVM	0.21	0.21	0.14	0.142	0.188	0.204
M	BoosTexter	-0.18	-0.01	-0.15	0.232	0.953	0.150
	IG	0.04	0.35	0.16	0.787	0.017	0.128
	Oddsratio	0.09	0.23	0.11	0.550	0.131	0.308
	Relief	0.10	0.10	0.18	0.512	0.500	0.088
	Ripper	0.08	-0.13	0.10	0.589	0.405	0.367
	SVM	0.01	-0.01	0.19	0.969	0.953	0.070
EvdS	BoosTexter	-0.01	0.01	0.22	0.944	0.985	0.032
	IG	0.13	0.31	0.27	0.373	0.037	0.008
	Oddsratio	0.13	-0.05	-0.02	0.354	0.728	0.817
	Relief	0.15	0.14	0.21	0.303	0.364	0.040
	Ripper	0.29	0.16	0.18	0.038	0.288	0.081
	SVM	0.15	-0.17	0.03	0.297	0.239	0.787
AK	BoosTexter	-0.05	0.24	0.13	0.721	0.108	0.189
	IG	0.36	-0.09	0.22	0.007	0.565	0.033
	Oddsratio	0.02	0.18	0.14	0.909	0.218	0.176
	Relief	0.26	0.26	0.26	0.055	0.084	0.009
	Ripper	0.00	0.22	0.23	0.987	0.142	0.024
	SVM	0.40	0.26	0.28	0.003	0.077	0.005
WS	BoosTexter	-0.17	-0.11	-0.12	0.226	0.461	0.235
	IG	-0.07	0.18	0.17	0.647	0.212	0.103
	Oddsratio	0.20	0.09	0.07	0.163	0.570	0.483
	Relief	0.22	0.32	0.23	0.133	0.030	0.028
	Ripper	0.28	0.07	0.20	0.048	0.623	0.060
	SVM	-0.23	0.0	0.04	0.106	0.623	0.722
GH	BoosTexter	0.20	0.20	0.27	0.199	0.183	0.012
	IG	0.02	0.13	0.22	0.896	0.389	0.039
	Oddsratio	0.35	0.19	0.14	0.023	0.196	0.180

	Relief	0.17	-0.14	0.09	0.257	0.358	0.387
	Ripper	0.06	0.27	0.25	0.727	0.072	0.018
	SVM	-0.14	0.15	-0.02	0.349	0.300	0.894
RV	BoosTexter	-0.10	-0.10	0.00	0.507	0.503	0.993
	IG	0.38	-0.02	0.10	0.008	0.928	0.347
	Oddsratio	0.05	0.12	0.17	0.755	0.406	0.091
	Relief	0.09	0.29	0.29	0.532	0.043	0.005
	Ripper	0.46	0.01	0.13	0.001	0.957	0.213
	SVM	0.20	-0.05	0.10	0.165	0.718	0.342
Mv B	BoosTexter	0.32	0.17	0.35	0.033	0.263	0.001
	IG	0.11	-0.10	-0.04	0.486	0.536	0.718
	Oddsratio	0.35	0.24	0.22	0.020	0.113	0.046
	Relief	-0.17	-0.06	-0.06	0.273	0.696	0.573
	Ripper	0.12	0.09	0.05	0.463	0.571	0.644
	SVM	-0.14	-0.14	-0.16	0.364	0.345	0.146
GW	BoosTexter	0.34	0.52	0.27	0.010	0.00	0.008
	IG	-0.01	-0.15	0.05	0.939	0.302	0.637
	Oddsratio	0.23	-0.13	0.00	0.090	0.369	0.975
	Relief	0.09	0.06	0.13	0.501	0.688	0.207
	Ripper	0.19	0.15	0.12	0.160	0.311	0.250
	SVM	-0.06	0.21	0.17	0.647	0.157	0.100

Table 16: Kendall's correlation coefficient for 12 entities

Reading the results from Table 16 is not very easy, that is why an overview is given in Figure 18. This figure provides for each feature selection technique and for each type of word the number of entities that had a significant positive correlation between the words selected by humans and the ones selected by the techniques. We only looked at positive correlation coefficients and not at the negative ones, as we only want humans and techniques to agree on each other. As can be seen from Figure 18, there are some entities where there was a positive correlation between words selected by humans and the ones selected by the techniques. The Ripper algorithm provides a clear positive correlation when considering only single words, i.e., there was for 5 of the 12 entities a correlation between the words selected by this algorithm and the words chosen by humans. Regrettably, this algorithm does not provide the same correlation for two consecutive words and moreover for composed words. The SVM ans the Relief algorithms performed very poor; for only 1 entity there was a significant correlation between the single words selected by this algorithm and the single words selected by humans. Oddsratio and BoosTexter are followed after the SVM algorithm when considering poor correlation for single words. There was no correlation found between the two consecutive words selected by Oddsratio, Ripper, and SVM and the words selected by the persons. The IG seems to perform in almost the same way for single, two consecutive, and composed words. Judging from the results in Figure 18 we see that BoosTexter and Relief perform quite well for composed words, i.e., there was for 6 of the 12 entities a correlation between the words selected by these algorithms and the words chosen by humans. So, ordening the techniques from best to worst, we get BoosTexter and Relief on the first place, followed by IG on the second place, Ripper, Oddsratio, and SVM.

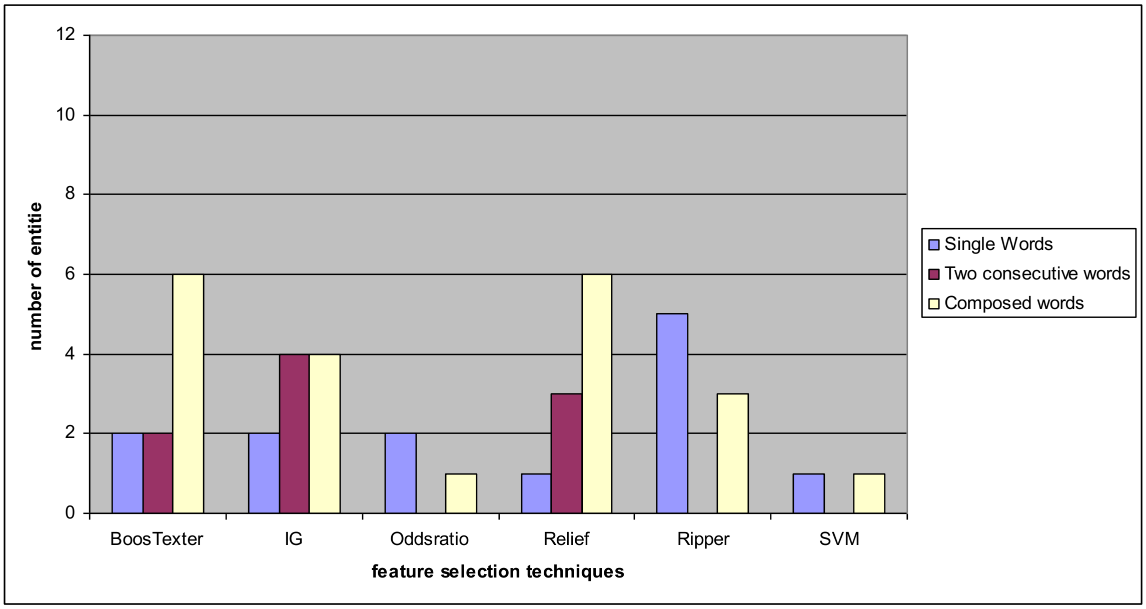

Figure 18: Number of times a significant positive correlation was found between a technique and humans

If we look at how the correlation coefficients of humans mutually are related to the correlation coeffients between words selected by humans and techniques, we notice (see Figure 19) that only in a few cases the correlation between humans is smaller than the correlation between humans and techniques. For example, for the entity AK the correlation between humans and the IG algorithm was 0.36 for single words (see table Table 16), where the correlation between humans mutually was 0.27 (see Table 14). This means that the IG algorithm and humans agree more on the selection of words than the humans mutually. The number of entities where a technique had a higher correlation coefficient compared to humans mutually, was maximal 2. Oddsratio was the worst technique, followed by SVM. The IG, Ripper, and SVM algorithms are doing well when considering single words. If we consider two consecutive words, we see that BoosTexter, IG and Relief are performing well. However when we look at composed words, we notice that only BoosTexter and Ripper agree more on the selection of words than the humans mutually.

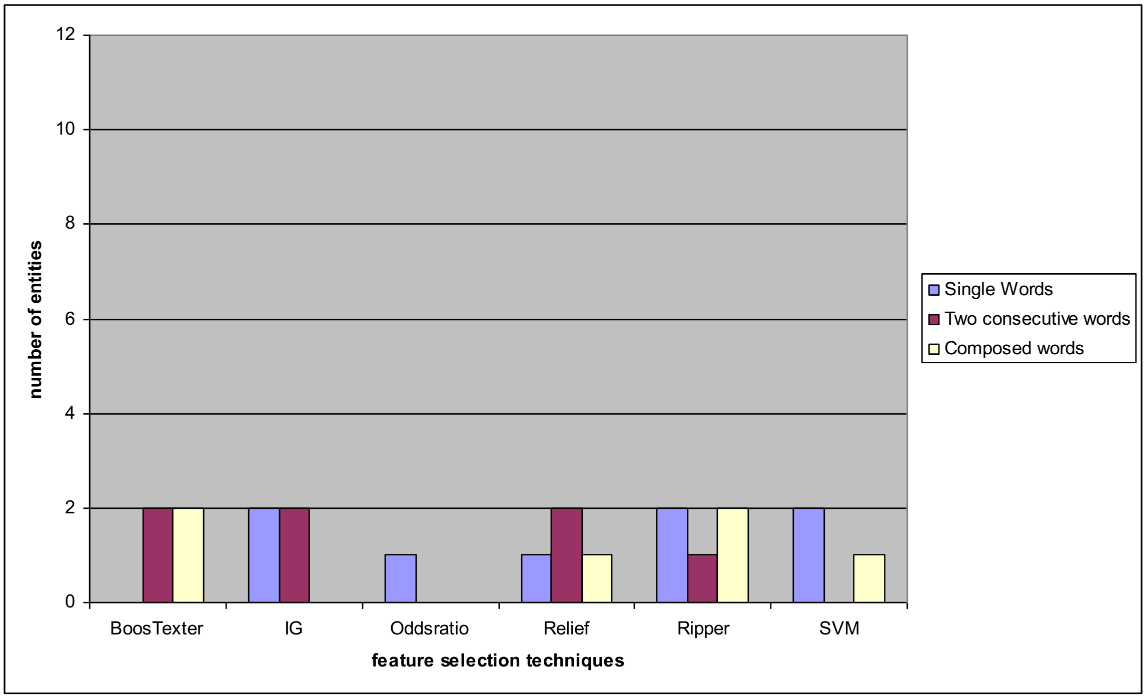

Figure 19: Number of times that the Kendall's correlation coefficient between techniques and humans was higher than the average correlation between humans

Note that next to calculating the correlation with Kendall's coefficient, we also compute the correlation with Spearman's coefficient. The results are provided in Appendix F. In general, the results from these two methods were not so different, that's why only one was chosen. Kendall's correlation coefficient was reported, because it is easier to interpret when the null hypothesis is rejected **[41]**.

8.3 Results of applying another type of editing

We are wondering whether the procedure of the composed words is good. One way to check this, is by merging the 10 single words and the 10 two consecutive words. We now obtain 20 composed words. It is interesting to see whether this simple merge procedure will lead to better correlations between humans and techniques for the composed words. The results for the composed words are given in Figure 20 (details can be found in Appendix H). From this Figure we can see that all the 6 feature selection techniques are performing more or less equally. Comparing these results with the one of Figure 18 we can conclude that if we do not perform any editing the results for the composed words will get worse. So, applying editing on the composed words is necessary.

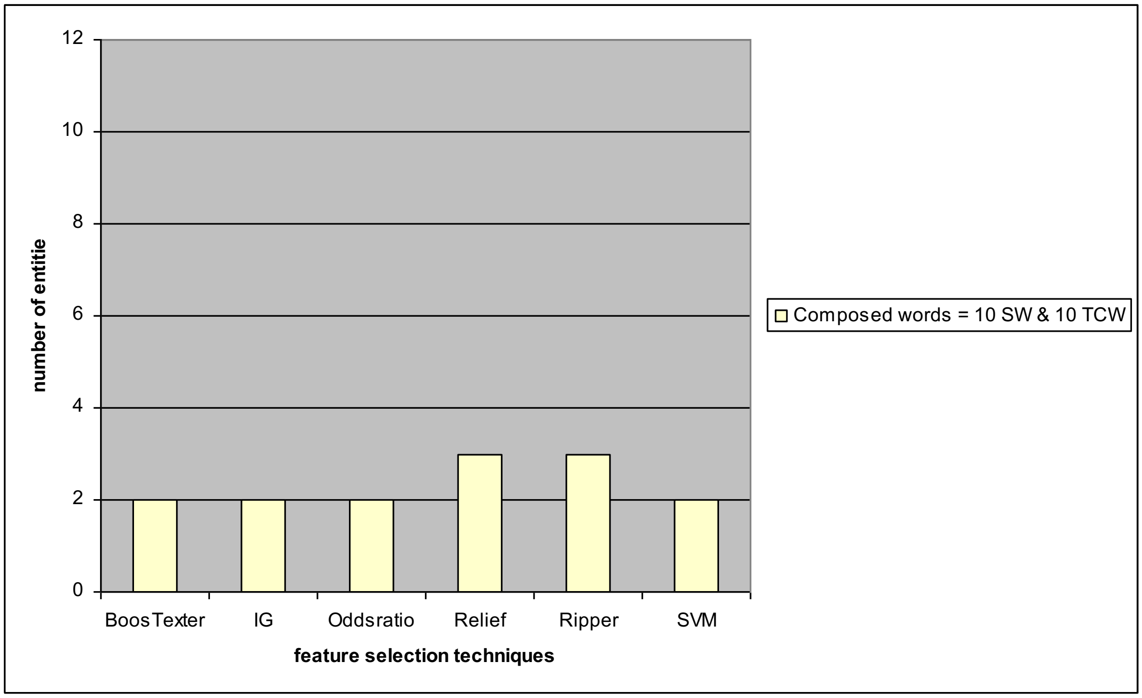

Figure 20: Number of times a significant positive correlation was found between a technique and humans for composed words

In our original set-up the composed words were basically generated by taking the top 5 single and two consecutive words, with taking the two consecutive word combination over a single word if the single word was part of the two consecutive words. We will now look if we can perform another form of editing, namely by assigning world knowledge to the words. Assign words that refer to the same person or object to the same class. We will explain it with an example. Suppose you have the words "benji", "madden" and "benji_madden". These three words refer to the same person, namely "Benji Madden". We will therefore sum the scores assigned by humans and provide a new label to these words, for example class_benji. If the total scores (of the humans) of the three words were respectively 1, 1, 3, then the score of class_benji will become 5. If the words benji and madden are ranked on the 1st and 2nd place for single words and the word benji_madden is ranked on the 1st place for two consecutive words (see Appendix A, entity PH, technique IG) then the score for the class_benji will be 10+9+10=29 for the composed words, 19 for single words, and 10 for two consecutive words. So, scores are grouped for both the words selected by the techniques and the words selected by the humans. The words that are grouped are given in Table 17 . As we are performing a form of editing we are taking those composed words that consist of the 10 single words and the 10 consecutive words. Thus, not the composed words from the original set-up.

Class type	Words		
class_vriend	vriend	vriendje	
class_music	heineken_music	music_hall	
class_lourdes	douchter_lourdes	lourdes	
class_fedde	fedde_le	le_grand	
class_mccartney	paul_mccartney	mccartney	
class_keeper	keeper	doelman	
class_oranje	nederlands_elftal	oranje	
class_ek	ek	europees_kampioenschap	
class_arsjavin	andrei_arsjavin	arsjavin	
class_coach	coach	bonscoach	
class_halvefinale	halve_finale	halve_finales	
class_melchiot	mario_mechiot	melchiot	
class_wilders	wilders	geert	geert_wilders
class_fitna	film	fitna	film_fitna
class_donor	orgaandonatie	orgaandonor	donor
class_nicole	nicole_richie	richie	nicole
class_benji	benji	madden	benji_madden
class_federline	kevin	federline	kevin_federline
class_lynn	jamie_lynn	lynn	jamie
class_guy	guy_ritchie	ritchie	guy
class_readmadrid	real	madrid	read_madrid
class_hiddink	guus	guus_hiddink	hiddink
class_russischeploeg	russische_ploeg	russische_spelers	russische_voetbal
	russische_elftal	russische_voetballers	russische_voetbalelftal

Table 17: Words that belong to the same class

Note that we only assigned classes to those words that were selected by humans. For example, the words lindsay and lohan were never selected, so we did not group the words lindsay , lohan, and lindsay_lohan. In theory this would make no difference. However, in practice it would save us some time.

As usual an overview is given for the techniques that had a positive significant Kendall's correlation coeffient (see Figure 21). So, these results are not improving when applying grouping. One of the reasons that the result are getting worser can be that a technique would only choose one of the three words ("benji", "madden", "benji_madden") where persons would select all the words or just the other way around.

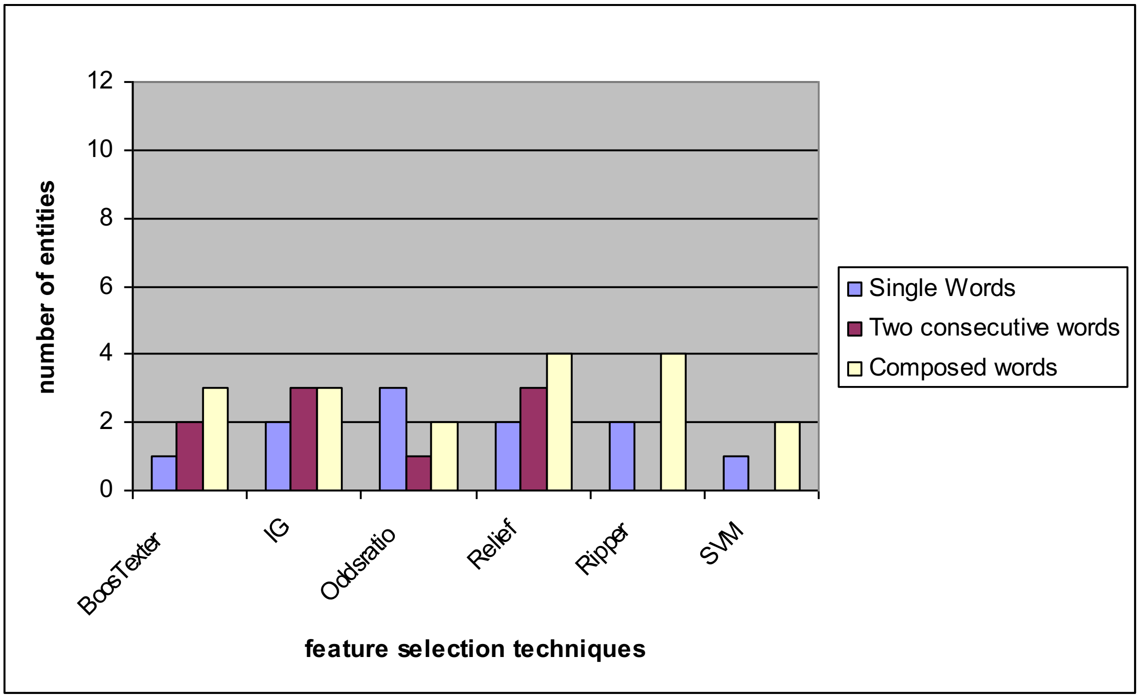

Figure 21: Number of times a significant positive correlation was found between a technique and humans

Based on these results we can conclude that we need to have an editing step. The reason why the results from Figure 21 are not better than the ones from Figure 18 could lie in the fact that we did not group all the words. For example, words like puppy and kopen and the word puppy_kopen are not assigned to one class. The idea is that if we would apply this kind of grouping the results would more or less be the same as the one in Figure 18. If we compare the results of Figure 21 with the results of Figure 20 for composed words, we see that the results of applying an editing on the composed words are slightly better than those without any editng.
Putting together all the results of Figure 18, Figure 20, and Figure 21, we observe the following. One, there should be definitely some editing done on the composed words. Two, editing the composed words by applying the world knowdlegde is more or less the same then when applying no knowledge at all but only a simple rule: take the top 5 single and two consecutive words, with first taking those two consecutive words where there exists a single word that is part of the two consecutive words. So, if there exists a word "benji" in the single words (regarless of its position in the list) and there exists a word "benji_madden" in the two consecutive words list (regarless of its position in the list), then the word "benji_madden" is taken.

The reason why there was in some cases no correlation found between humans and techniques, is probably because each test person has a different opinion about an entity. Meaning, based on a test person background the words are selected. Also, both single and two consecutive words are provided at once. This made it very hard for test persons to choose. For example, the words "ek" and "europees_kampioenschap" or the words "doelman" and "keeper". These 2 words mean exactly the same, so choosing between these 2 can be very difficult. Perhaps the experiment should be improved: give persons a list that contains only single words, a list of only two consecutive words, and a list of only composed words. For each list they should then select the 10 and 5 most characteric words. So, we would contain 3 lists instead of 1. We would expect to achieve a better correlation between humans and techniques with this set-up. Because of time constaints this was not done and also because test persons where complaining about how much time it took to select words in for 12 entities. If they would get 3 lists instead of 1, it could then probably result in getting no results at all.

8.4 Nominal concordance

Besides, calculating few, representative, distinctive, and relevant words, it was also interesting to see which feature selection technique was the most stable one. This stability can be measured by the nominal concordance. The nominal concordance for four entities-techniques for single, two consecutive, and composed words are provided in Table 18.

Entity	Technique	Nominal concordance		
		Single words	Two consecutive words	Composed words
PH	BoosTexter	0.69	0.72	0.72
	IG	0.76	0.85	0.71
	Oddsratio	1	1	1
	Relief	0.92	0.71	0.76
	Ripper	0.53	0.79	0.72
	SVM	0.5	0.69	0.62
SD	BoosTexter	0.66	0.67	0.56
	IG	0.36	0.64	0.67
	Oddsratio	1	1	1
	Relief	0.87	0.52	0.70
	Ripper	0.24	0.75	0.52
	SVM	0.23	0.54	0.32
BS	BoosTexter	0.71	0.87	0.83
	IG	0.92	0.86	1
	Oddsratio	1	1	1
	Relief	0.86	0.86	0.78
	Ripper	0.67	0.81	0.68
	SVM	0.65	0.70	0.67
AA	BoosTexter	0.69	0.79	0.68
	IG	0.85	0.90	0.93
	Oddsratio	1	1	1
	Relief	0.90	0.82	0.84
	Ripper	0.61	1	0.83
	SVM	0.56	0.54	0.66
AK	BoosTexter	0.75	0.80	0.90
	IG	0.91	0.86	0.93

	Oddsratio	1	1	1
	Relief	0.96	0.93	0.90
	Ripper	0.65	0.76	0.81
	SVM	0.54	0.51	0.44

Table 18: Nominal concordance for 5 entities

From Table 18 and Figure 22 it is obvious that Oddsratio is the most stable technique. The nominal concordance of Oddsratio is not only one for all the five entities, but also for all the word types (single, two consecutive, composed words). This means that it does not matter how many random samples one takes, the words that are selected by odds ratio are always the same. This is convenient, since it saves a lot of computational time. The technique that is less stable is SVM followed by Ripper.

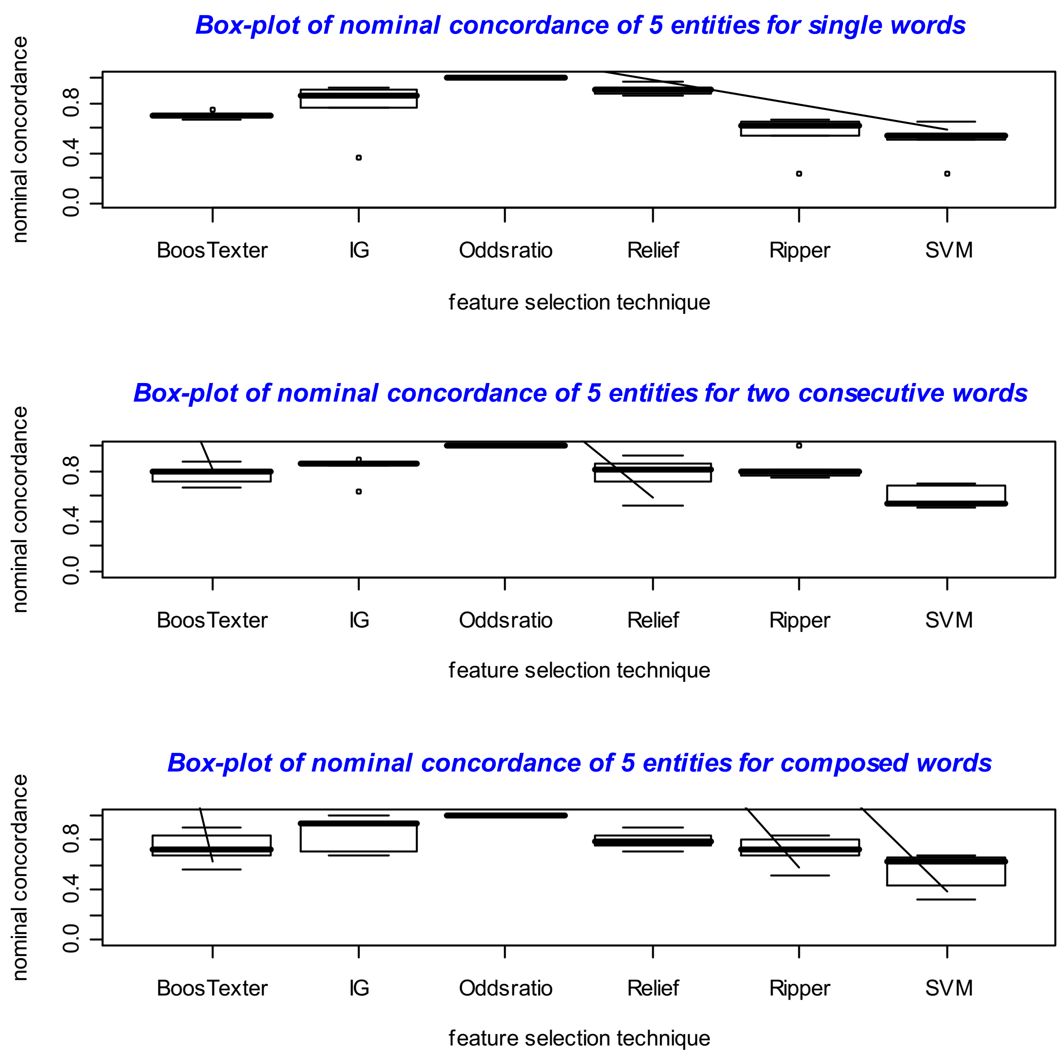

Figure 22: Box-plots of nominal concordance SW, TCW, and CW

8.5 Computational time

One of the probably most important things to know is how much time was required to achieve the results. The time required to do a single cross validation fold is given in Table 19. In this table the mimimum and maximum time required for all the 12 entities for one single cross validation fold is provided.

Technique	Time to do a single cross validation fold	
	Single words	Two consecutive words
BoosTexter	8 seconds – 1 minute and 53 seconds	6 seconds – 1 minute and 28 seconds
IG	0 seconds – 4 seconds	0 seconds – 3 seconds
Oddsratio	0 seconds – 1 second	0 seconds – 1 second
Relief	0 seconds – 10 minutes and 46 seconds	1 second – 10 minutes and 44 seconds
Ripper	1 seconds – 1 minute and 35 seconds	1 second – 1 minute and 29 seconds
SVM	4 seconds – 4 hours and 29 minutes	1 second – 5 hours and 5 minutes

Table 19: Time that could be required for a random entity

Technique	Approximate time complexity
BoosTexter	$O(D * V * T)$
IG	$O(D * V)$
Oddsratio	$O(D * V)$
Relief	$O(D * V * n)$
Ripper	$O(D * \log^2(D))$
SVM	$O(V * D^{1.7})$

Table 20: Approximate time complexity for each technique

From Table 19 we can see that the time required to do single words and two consecutive words is almost the same. The CPU time for each entity for single words is illustrated with a graph (see Figure 23). From this figure it is obvious that SVM is the only technique that requires an extreme large computational time. In order to get a better picture for the rest of the techniques, we will take out the SVM (see Figure 24). From Figure 24 and Table 19 we can see that Oddsratio and IG are the fastest techniques, followed by Ripper and BoosTexter. We can observe that the time required for Ripper and BoosTexter to select words does not differ very much. The Relief algorithm followed on the fifth place. Also, observe that the time complexity given in Table 20 is consistent with our results of the time we found for the entities. Note that the preprocessing step for Oddsratio and BoosTexter is done in Perl. The CPU time required for this step is not included.

Furthermore keep in mind that BoosTexter is implemented in C++, Oddratio in Perl, while the rest of the methods are implemented in Java.

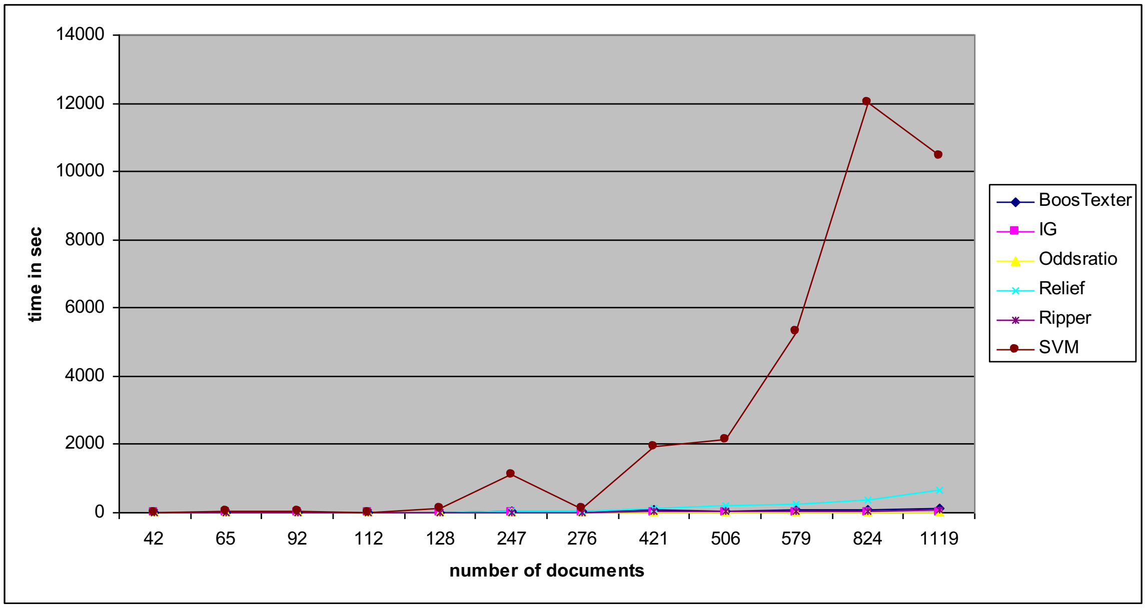

Figure 23: CPU time required for a single cross validation fold (all techniques)

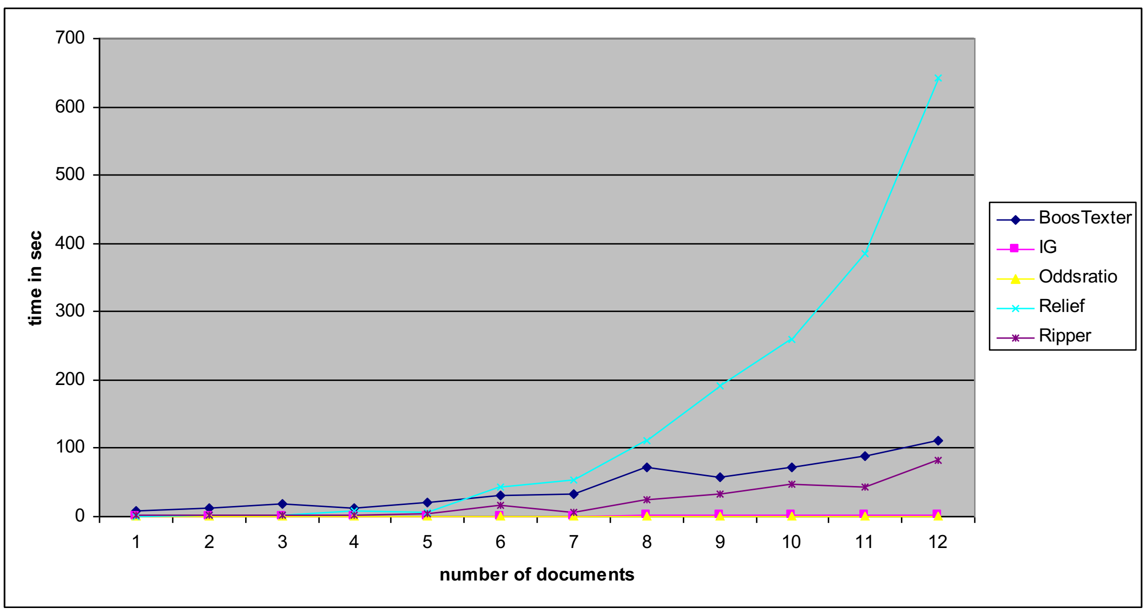

Figure 24: CPU time required for a single cross validation fold (all techniques except SVM)

9 Conclusion and recommendations

The purpose of this study was to find a technique that is able to generate a high quality profile for an entity. This profile should contain not only few, but also representative, distinctive, and relevant words. Different data mining techniques were tried to solve this problem. To be precise, the techniques Oddsratio, Information Gain, Ripper, Relief, SVM-FS, and BoosTexter were applied. The distinctiveness and the representativeness of these words were evaluated by evaluating the performance of a machine learning algorithm that is based on only these words as features. SVM-Class was used as classifier and the performance was measured by calculating the F_1-measure. There were 12 entities considered and for each entity-technique the F_1-measure was calculated for single, two consecutive, and composed words. It turned out that there was no significant difference between the techniques when looking at single and composed words. For two consecutive words the Relief algorithm was the one that was creating a significant difference, meaning that there was no significant difference when looking at the F_1-measure of the rest of the techniques (Oddsratio, Information Gain, Ripper, SVM-FS, and BoosTexter). As we are more interested in the composed words, because not all words can be described in single words, and on the other hand two consecutive words are not always enough to describe a word. For example, if we would consider a word like London. This word can never be described in a two consecutive word combination. However, if we consider a word like New York, then this word can also never be captured in a single word. Therefore we need to have composed words. As for the composed words there was no significant difference when looking at the F_1-measure, we can basically conclude that each technique is suited for generating a profile containing distinct and representative words.

We did not only compute the F_1-measure, but also the correlation between the words selected by humans and the techniques. This was necessary in order to deterime which technique was able to produce relevant (and representative) words.We used Kendall's correlation coefficient to determine the correlation and also looked at the p-value to determine if this correlation coefficient was significant, i.e., can we reject the null hypothesis and assume that there is a positive correlation? It turned out that when looking at single words, the Ripper algorithm performed best; words selected from 5 of the 12 entities with this algorithm had a positive correlation with words selected by humans. For two consecutive words only Information Gain had 4 out of the 12 entities that had a positive correlation followed by Relief which had 3 out of the 12. Note that for us the most important result is the one obtained for composed words. The best result yielded 6 out of the 12 entities. This positive correlation between words produced by humans and techniques was achieved by BoosTexter and Relief. For 4 out of the 12 entities there was positive correlation between words produced by humans and the Information Gain algorithm. SVM together with Oddsratio and Ripper performed worse.

If we would consider the time required to do one single cross validation, we can conclude that Oddsratio and Information Gain are the fastest techniques, followed by BoosTexter and Ripper. On the 5th place ends the Relief algorithm followed by the SVM technique, which is the only technique that requires an extremely huge amount of time.

We were not only interested in which technique was able to generate few, representative, distinct, and relevant words, but also in the stability of each technique, i.e., which technique was able to generate the same words given different negative documents. This stability was measured by computing the nominal concordance. The nominal concordance was computed for only 5 entities since only for these entities the number of random samples was at least 5. It turned out to be that Oddsratio was the most stable technique among the others. Regardless which type of word we considered (single, two consecutive, or composed words) the nominal concordance was always one.

As our main goal was to find a technique that is able to produce a compressed and high quality profile, we can more or less do not take into account the results found for the stability for now. Basically any technique can be chosen based on the results of the F_1-measure. However, if we consider the correlation we notice that there is a weak correlation between techniques and humans, and therefore we cannot choose any technique we want. The BoosTexter algorithm is preferred over the Relief alogorithm, because the last one takes more time to select the words when the dataset is increasing. Choosing between BoosTexter and the Information Gain algorithm is not easy, because BoosTexter on one hand performed slightly better than Information Gain. Information Gain on the other hand takes less time to select the words, i.e., the time for BoosTexter to generate the words can be up to 2 minutes where the for Information Gain to generate the words can be up to 4 seconds (for one single cross validation fold). So, if one wants a technique that is fast and qualitatively not superior, one can choose Information Gain. On the other hand if one wants a technique that produces qualitatively better results and one has time enough, then one can go for BoosTexter. Also, Information Gain is easier to understand than BoosTexter.

9.1 Future work

We only looked at single and two consecutive words, and a combination of these two. However, this does not cover all the words. Some words, such as ‘heineken music hall’, ‘trots op nederland’, ‘fedde le grand’, consist of more than two consecutive words. Therefore it would be interesting to look at more consecutive words.

As one can see from Appendix A not all words are meaningfull. In other words, we would like to have some words filtered before applying any feature selection technique. A simple example is a word like ‘zahra90gestoorde_meiddoukaliaatj__sphfemocpowvovgn0m0red0ubtfemmefatalemarokkaantjuhbanditanieuw_middagnieuwsbriefwenen_duitsland’. One way to solve this problem is to exclude words that are longer than a certain threshold. Another way is to implement an advanced tool that is able to filter all the unneccassary words, tabs, etcetera of a text such that it in the end only contains the actual text. It is important to clean the data, if one wants to continue this work. Also, if a word contains a special character such as ‘ë’, ‘è’, ‘ï’, these characters are lost. A simple example is the word ‘financiële’. From this word the ‘ë’ was lost. This issue is probably the only one that can be solved easily, namely by changing the locale settings in Perl. Another thing that needs to be done is filter out words that more or less mean the same or refer to the same thing. For example, if we consider words like ‘nederlands elftal’ and ‘nederlandse ploeg’ or words like ‘nederland’ and ‘oranje’, then we see that these words refer to the same thing. Also, words that mean the same but are provided in different languages needs also to be taken out. In this project words like ‘keeper’ and ‘doelman’ appeared. Both words mean the same and are only written in different languages, namely English and Dutch. Next to these words, there are also words that contain spelling differences, such as ‘andrei arsjavin’ and ‘andrei arshavin’ or ‘dirk kuyt’ and ‘dirk kuijt’. These two words refer to the same person and are now only written differently.

Another thing that can also be done in the future is improving the experiment by providing three lists to persons (single words list, two consecutive words list, and composed words list) instead of 1 list.

10 Appendix

A Final words

This final list produced for single (SW), two consecutive (TCW), and composed words (CW) is provided.

Entity	Data set = 200 features			
Max Neg un = 5 x Max Sample d = 2 x Cross-validati on = 5x	Technique	SW	TWC	CW
Paris Hilton	BoosTexter	puppy madden moeder geld duidelijk blatt mtv beckinsale foto's sabine	kevin_blatt bloedeigen_parfumlijn benji_madden nicole_richie puppy_kopen raar_trekje joel_madden duikt_studio showbizz_sloerie jessica_batzers	kevin_blatt bloedeigen_parfumlijn moeder benji_madden nicole_richie puppy_kopen joel_madden duidelijk raar_trekje geld
	IG	madden benji richie nicole mtv vriendje feestjes harlow puppy joel	benji_madden nicole_richie beste_vriendin kevin_blatt simple_life new_bff vriendje_benji joel_madden love_guru my_new	benji_madden nicole_richie beste_vriendin kevin_blatt mtv vriendje_benji new_bff love_guru simple_life joel_madden
	Oddsratio	amerikaanse benji madden euro beste dochter mtv kleine nicole new	benji_madden beste_vriendin amerikaanse_tijdschrift britney_spears love_guru los_angeles my_new nicole_richie new_bff joel_madden	benji_madden beste_vriendin amerikaanse_tijdschrift britney_spears amerikaanse my_new dochter mtv nicole_richie love_guru
	R	puppy	raar_trekje	raar_trekje

	elief	foto's kopen vriendje gespot geld studio zwanger vriend duidelijk	david_beckham nicole_richie miljoen_euro puppy_kopen los_angeles offici_le carri_re benji_madden new_york	foto's david_beckham vriendje miljoen_euro puppy_kopen gespot nicole_richie vriend offici_le
	Ripper	madden puppy bloedeigen amerikaanse nietsnut kevin blatt beckinsale kleine benji	benji_madden nicole_richie kevin_blatt puppy_kopen britney_spears euro_geboden jessica_batzers raar_trekje showbizz_sloerie bloedeigen_parfumlijn	benji_madden nicole_richie kevin_blatt bloedeigen_parfumlijn bloedeigen amerikaanse puppy_kopen britney_spears euro_geboden jessica_batzers
	SVM	puppy madden verne vriendje kim mtv beckinsale stopen parfumlijn lilliputter	raar_trekje benji_madden puppy_kopen kevin_blatt bloedeigen_parfumlijn nicole_richie verne_troyer joel_madden offici_le vriend_steven	sabine benji_madden puppy_kopen raar_trekje kevin_blatt bloedeigen_parfumlijn vriendje kim verne mtv
Snoop Dogg	BoosTexter	amsterdam olie druk jongeren binnenkort rechtbank ontvangen overwinning leuke dertig	ek_jan hoog_niveau druk_momenteel acteurs_zetten heineken_music londense_luchthaven nicolas_sarkozy europese_ministers eiland_aruba grootste_iraanse	amsterdam olie jongeren hoog_niveau ek_jan druk_momenteel binnenkort heineken_music londense_luchthaven rechtbank
	IG	daalde tomtom opnemen behalve vat welkom rapper olieprijs heineken hogere	heineken_music iraanse_bank grootste_iraanse music_hall amerikaanse_rapper nieuw_album marks_brengt maak_acteurs missy_elliott nicolas_sarkozy	heineken_music opnemen behalve olieprijs tomtom iraanse_bank grootste_iraanse music_hall daalde welkom
	Odds	amerikaanse amsterdam altijd goed nederland	heineken_music ge_nteresseerd georgina_verbaan music_hall grootste_iraanse	amsterdam altijd heineken_music amerikaanse ge_nteresseerd

	ratio	druk europese ek euro tweede	amerikaanse_rapper amy_winehouse iraanse_bank sylvie_viel ballistische_raketten	georgina_verbaan grootste_iraanse amerikaanse_rapper amy_winehouse goed
	Relief	amsterdam binnenkort druk com rijden leuke vrij iemand allemaal olie	frans_bauer miljoen_euro jan_smit thomas_berge verenigde_staten new_york gerard_joling tweede_kamer georgina_verbaan iraanse_bank	frans_bauer halve_finale amsterdam jan_smit miljoen_euro binnenkort new_york thomas_berge verenigde_staten druk
	Ripper	ontvangen gisteren amsterdam iran music rapper amerikaanse opnemen jongen overwinning	grootste_iraanse heineken_music aandeel_noteerde acteurs_zetten john_marks amerikaanse_ministerie druk_momenteel hoog_niveau amerikaanse_rapper eiland_aruba	grootste_iraanse heineken_music aandeel_noteerde acteurs_zetten ontvangen gisteren amsterdam iran amerikaanse_ministerie amerikaanse_zakenbank
	SVM	tomtom iran waaronder binnenkort heineken welkom dertig opnemen hogere rapper	music_hall vari_rend overbelast_raakt tu_delft marks_brengt hoog_niveau tori_spelling zahra90gestoorde_meiddoukaliaatj_ _sphfemocpowvovgn0m0red0ubtfe mmefatalemarokkaantjuhbandita nieuw_middagnieuwsbrief wenen_duitsland	jongen tomtom music_hall vari_rend overbelast_raakt tu_delft marks_brengt hoog_niveau iran waaronder
Britney Spears	BoosTexter	angeles kevin lynn federline dolls foto's album 26-jarige jamie lindsay	kevin_federline jamie_lynn lindsay_lohan pussycat_dolls los_angeles nieuw_album beste_artiest 26-jarige_popidool ok_magazine voorprogramma_kane	kevin_federline jamie_lynn lindsay_lohan pussycat_dolls los_angeles foto's nieuw_album beste_artiest zangeres 26-jarige_popidool
	IG	dolls pussycat kevin jamie federline zangeres 26-jarige	pussycat_dolls kevin_federline jamie_lynn clip_pussycat sean_preston jayden_james los_angeles	pussycat_dolls kevin_federline jamie_lynn clip_pussycat jayden_james los_angeles sean_preston

		clip lynn emmy	lindsay_lohan maddie_briann tijdschrift_people	zangeres 26-jarige maddie_briann
	Oddsratio	amerikaanse 26-jarige dolls angeles kevin clip jamie los federline pussycat	amerikaanse_tijdschrift clip_pussycat jamie_lynn kevin_federline los_angeles jayden_james lindsay_lohan pussycat_dolls maddie_briann sean_preston	amerikaanse_tijdschrift clip_pussycat 26-jarige jamie_lynn kevin_federline los_angeles jayden_james lindsay_lohan pussycat_dolls grow_up
	Relief	foto's nederland album nieuw kinderen drank goed echt rechter angeles	pussycat_dolls miljoen_euro kevin_federline sean_preston jayden_james nieuw_album jamie_lynn los_angeles grow_up paris_hilton	pussycat_dolls miljoen_euro foto's kevin_federline gerard_joling nederland nieuw_album sean_preston los_angeles kinderen
	Ripper	dolls kevin zangeres lindsay 26-jarige emmy federline jamie bekend angeles	pussycat_dolls jamie_lynn kevin_federline lindsay_lohan amerikaanse_tv-prijs 26-jarige_popidool beste_artiest mel_gibson voorprogramma_kane los_angeles	pussycat_dolls jamie_lynn kevin_federline lindsay_lohan zangeres amerikaanse_tv-prijs 26-jarige_popidool emmy beste_artiest bekend
	SVM	lynn jamie pussycat kevin overrijden lohan emmy lindsay federline 26-jarige	pussycat_dolls lindsay_lohan kevin_federline voorprogramma_kane jamie_lynn mel_gibson beste_artiest you_mother studio_ingedoken zangeres_haalt	pussycat_dolls lindsay_lohan kevin_federline voorprogramma_kane jamie_lynn overrijden emmy mel_gibson beste_artiest you_mother
Ahmed Aboutaleb	BoosTexter	sociale staatssecretaris bericht amsterdamse ministerie sowieso actie verwacht ontwikkeling goed	islamitische_scholen inkomen_cwi sociale_zaken zaken_werkt automatisch_kwijtschelding arme_kinderen arme_gezinnen aow_uitvoert kerken_vorig anderhalf_miljoen	islamitische_scholen staatssecretaris inkomen_cwi sociale_zaken bericht zaken_werkt automatisch_kwijtschelding ontwikkeling ministerie amsterdamse

	IG	sociale staatssecretaris zaken wetsvoorstel gemeentelijke gemeenten binnenkort gezinnen utrecht uitkering	sociale_zaken zaken_werkt arme_kinderen honderd_gemeenten arme_gezinnen extra_ondersteuning totaal_kinderen overeenkomst_ondertekend sociale_verzekeringsbank lokale_belastingen	sociale_zaken zaken_werkt arme_kinderen staatssecretaris wetsvoorstel honderd_gemeenten arme_gezinnen gemeentelijke extra_ondersteuning utrecht
	Oddsratio	gemeenten den euro miljoen nederland haag sociale alleen staatssecretaris utrecht	arme_gezinnen arme_kinderen den_haag extra_ondersteuning honderd_gemeenten miljoen_euro sociale_zaken aow_uitvoert voorpagina_binnenland lokale_belastingen	arme_gezinnen arme_kinderen den_haag honderd_gemeenten nederland miljoen_euro extra_ondersteuning alleen sociale_zaken staatssecretaris
	Relief	sociale staatssecretaris zaken ministerie bedrijven nederlanders onderwijs kinderen euro bericht	sociale_zaken den_haag tweede_kamer geert_wilders buitenlandse_zaken miljoen_euro zaken_werkt andr_rouvoet arme_kinderen extra_ondersteuning	sociale_zaken den_haag staatssecretaris tweede_kamer geert_wilders ministerie buitenlandse_zaken miljoen_euro andr_rouvoet nederlanders
	Ripper	staatssecretaris sociale amsterdamse helpen ministerie bedrijven kinderen onderwijs ontwikkeling inkomen	sociale_zaken islamitische_scholen bovendien_beschikken automatisch_kwijtschelding den_haag zaken_werkt arme_kinderen	staatssecretaris sociale_zaken islamitische_scholen bovendien_beschikken automatisch_kwijtschelding helpen ministerie amsterdamse bedrijven kinderen
	SVM	sociale staatssecretaris wetsvoorstel sowieso centrum caf buitenlandse kabinet ontwikkeling amsterdamse	sociale_zaken islamitische_scholen twaalf_maanden gemeentelijke_belastingen automatisch_kwijtschelding openbare_scholen nederland_ruim den_haag verzekeringsbank_svb arme_kinderen	sociale_zaken islamitische_scholen staatssecretaris cwi twaalf_maanden gemeentelijke_belastingen wetsvoorstel centrum sowieso zaken_werkt
Madonna	BoosT	echt man procent president album landen	fedde_le amy_macdonald frank_lammers alex_klaasen do_vrij verenigde_staten	echt fedde_le amy_macdonald man frank_lammers alex_klaasen

	exter	partijen contact partij love	guy_ritchie billie_holiday dima_bilan album_top	procent president do_vrij dima_bilan
	IG	guy ritchie minutes album life fedde nelly amy grand lourdes	guy_ritchie fedde_le amy_macdonald le_grand mega_charts gfk_mega dochter_lourdes dima_bilan artiest_nummer hard_candy	guy_ritchie minutes fedde_le amy_macdonald le_grand life gfk_mega mega_charts dochter_lourdes album
	Oddsratio	album alleen guy nederlandse echt nederland nummer amy amsterdam huwelijk	amy_macdonald artiest_nummer fedde_le guy_ritchie dochter_lourdes gfk_mega le_grand amy_winehouse dima_bilan mega_charts	amy_macdonald album alleen artiest_nummer guy_ritchie fedde_le echt dochter_lourdes gfk_mega nederland
	Relief	trouwring guy amsterdam contact love live echt nederland man oranje	britney_spears den_haag miljoen_euro guy_ritchie wolter_kroes gerard_joling georgina_verbaan frans_bauer new_york fedde_le	britney_spears den_haag miljoen_euro trouwring guy_ritchie amsterdam wolter_kroes contact love live
	Ripper	album guy life minutes tournee top schmidt nelly fedde amy	guy_ritchie fedde_le amy_macdonald alex_klaasen dima_bilan frank_lammers buitenlandse_zaken balkenende_cda mexico_city do_vrij	guy_ritchie fedde_le amy_macdonald alex_klaasen minutes life top tournee dima_bilan buitenlandse_zaken
	SVM	nelly trouwring remix guy mccartney schmidt minutes sticky kraaijkamp life	sticky_sweet guy_ritchie oude_fiets paul_mccartney mexico_city frank_lammers vrij_za engelstalige_album nelly_furtado balkenende_cda	sticky_sweet guy_ritchie tournee trouwring remix schmidt nelly oude_fiets paul_mccartney mexico_city

Edwin van der Sar	BoosTexter	wk goed groot nederland elftal zit toernooi oranje kreeg kwartfinale	guus_hiddink wesley_sneijder manchester_united dennis_bergkamp champions_league andr_ooijer orlando_engelaar europees_kampioen arjen_robben groot_toernooi	guus_hiddink wesley_sneijder oranje manchester_united wk dennis_bergkamp aanvoerder goed champions_league andr_ooijer
	IG	doelman recordinternati onal oranje nederlands bronckhorst giovanni elftal andr ruud ek	nederlands_elftal wesley_sneijder andr_ooijer dennis_bergkamp manchester_united nederlandse_toernooispeler meest_ervaren ervaren_nederlandse toernooispeler_aller dirk_kuijt	doelman nederlands_elftal wesley_sneijder andr_ooijer recordinternational oranje dennis_bergkamp manchester_united meest_ervaren bronckhorst
	Oddsratio	basten doelman ek elftal nederland nederlands itali bondscoach oranje spelers	andr_ooijer arjen_robben bondscoach_marco dirk_kuijt guus_hiddink khalid_boulahrouz manchester_united nederlands_elftal europees_kampioenschap dennis_bergkamp	andr_ooijer basten arjen_robben doelman bondscoach_marco ek dirk_kuijt nederland itali nederlands_elftal
	Relief	goed keeper ek nooit oranje goede binnen spanje meeste dagen	guus_hiddink nederlands_elftal bondscoach_marco komend_seizoen miljoen_euro ek_voetbal europees_kampioenschap europees_kampioen halve_finale real_madrid	goed guus_hiddink nederlands_elftal bondscoach_marco keeper komend_seizoen nooit oranje spanje ek_voetbal
	Ripper	doelman oranje nederlands vaart keeper elftal wesley spelers toernooi nederlandse	nederlands_elftal andr_ooijer wesley_sneijder meest_ervaren manchester_united aller_tijden dirk_kuijt gianluigi_buffon bondscoach_marco khalid_boulahrouz	nederlands_elftal doelman andr_ooijer oranje dennis_bergkamp wesley_sneijder meest_ervaren spelers dirk_kuijt vaart
	SV	recordinternati onal giovanni	warme_familie verloor_oranje olympisch_museum	warme_familie recordinternational verloor_oranje

	M	andr lat record bergkamp interland verdediging doelman kuijt	yuri_zhirkov phillip_cocu verdedigende_middenvelder meerdere_spelers vaart_binnen sidney_govou meest_ervaren	giovanni lat andr petr_cech yuri_zhirkov record verdedigende_middenvelder
Ab Klink	BoosTexter	volksgezondhe id minister rouvoet moest overleg erfelijke onderwerp horeca sprake cda	den_haag tweede_kamer pati_nten vicepremier_andr horeca_nederland kabinet_buigt pati_nt extra_ambulances nederlandse_zorgautoriteit dagen_stoppen	volksgezondheid den_haag tweede_kamer pati_nten rouvoet minister vicepremier_andr moest horeca_nederland overleg
	IG	volksgezondhe id minister orgaandonatie organen systeem donor automatisch orgaandonor nabestaanden bezwaar	automatisch_donor dood_organen horeca_nederland vicepremier_andr nederland_khn koninklijk_horeca pati_nten kabinet_buigt compromis_houdt co_rdinatiegroep	automatisch_donor volksgezondheid dood_organen minister orgaandonatie horeca_nederland vicepremier_andr systeem pati_nten nederland_khn
	Oddsratio	cda den haag kabinet brief minister kamer binnen nederland tweede	andr_rouvoet automatisch_donor den_haag bussemaker_pvda dood_organen jet_bussemaker academisch_ziekenhuis pati_nten tweede_kamer staatssecretaris_jet	andr_rouvoet cda automatisch_donor den_haag bussemaker_pvda kabinet brief academisch_ziekenhuis binnen dood_organen
	Relief	volksgezondhe id minister zorg ministerie nederland kamer tweede haag zorgverzekeraa rs land	wouter_bos tweede_kamerlid tweede_kamer miljoen_euro jan_peter pati_nten peter_balkenende den_haag premier_jan medisch_centrum	volksgezondheid wouter_bos minister tweede_kamerlid tweede_kamer zorg miljoen_euro ministerie jan_peter pati_nten
	Ri	volksgezondhe id minister	automatisch_donor vicepremier_andr horeca_nederland	volksgezondheid minister automatisch_donor

	pper	bussemaker ministers commissie rouvoet erfelijke meest mits ziekte	nederlandse_zorgautoriteit pati_nten kabinet_buigt extra_ambulances academisch_ziekenhuis andr_rouvoet compromis_houdt	horeca_nederland vicepremier_andr nederlandse_zorgautoriteit compromis_houdt pati_nten bussemaker extra_ambulances
	SVM	volksgezondheid minister orgaandonor zorgverzekeraars orgaandonatie jet toestemming bussemaker roken donor	zorgverzekeraars_vergoed overmatig_alcoholgebruik horeca_nederland uitgelekte_brief laatste_woord nederlandse_zorgautoriteit pati_nten automatisch_donor kabinet_buigt onomkeerbare_stappen	minister volksgezondheid zorgverzekeraars_vergoed overmatig_alcoholgebruik uitgelekte_brief orgaandonatie zorgverzekeraars toestemming pati_nten onomkeerbare_stappen
Wesley Sneijder	BoosTexter	arjen roemeni itali real madrid nederland elftal rafael tweede man	real_madrid dirk_kuijt arjen_robben wereldkampioen_itali europees_kampioen dirk_kuyt den_haag beste_speler joris_mathijsen ibrahim_afellay	tweede real_madrid dirk_kuijt arjen_robben wereldkampioen_itali kreeg man europees_kampioen den_haag dirk_kuyt
	IG	itali robben persie arjen dirk real robin madrid kuijt ruud	arjen_robben real_madrid dirk_kuijt nederlands_elftal beste_speler wedstrijd_verkozen gianluigi_buffon david_villa beide_oranje-internationals hamit_altintop	itali arjen_robben real_madrid persie dirk_kuijt beste_speler nederlands_elftal wedstrijd_verkozen robin david_villa
	Oddsratio	basten ek elftal itali nederland nederlands oranje persie robben wedstrijd	arjen_robben bondscoach_marco david_villa dirk_kuijt europees_kampioenschap nederlands_elftal khalid_boulahrouz orlando_engelaar real_madrid michael_ballack	arjen_robben basten bondscoach_marco ek david_villa dirk_kuijt nederland nederlands_elftal oranje wereldkampioen_itali
	Reli	ek bondscoach rusland hiddink	guus_hiddink nederlands_elftal bondscoach_marco afgelopen_seizoen	guus_hiddink nederlands_elftal bondscoach_marco rusland

	ef	basten guus marco oranje seizoen nederlands	fc_twente halve_finale europees_kampioenschap ek_voetbal khalid_boulahrouz arjen_robben	afgelopen_seizoen fc_twente basten halve_finale oranje ek_voetbal
	Ripper	itali middenvelder ruud robben robin rafael persie madrid speler dirk	arjen_robben dirk_kuijt real_madrid david_villa nederlands_elftal beide_oranje-internationals guus_hiddink michael_ballack nederland_mist gianluigi_buffon	itali arjen_robben middenvelder dirk_kuijt real_madrid ruud david_villa robin rafael nederlands_elftal
	SVM	dirk real madrid prachtige treffers guus giovanni roman engelaar orlando	vloek_ontzag tweede_treffer prachtige_aanval nederland_mist tweede_gele zeven_doelpunten oranje_discussie verschillende_spelers gianluigi_buffon individuele_kwaliteiten	vloek_ontzag wedstrijd_verkozen real tweede_treffer madrid prachtige_aanval engelaar nederland_mist treffers giovanni
Guus Hiddink	BoosTexter	bondscoach oranje wedstrijd ek russen team spanje basel russische ploeg	nederlands_elftal halve_finales ek_voetbal roman_pavljoetsjenko halve_finale europees_kampioen lagerb_ck khalid_boulahrouz andrei_arsjavin andrei_arshavin	nederlands_elftal bondscoach oranje ek_voetbal wedstrijd russen roman_pavljoetsjenko halve_finales team europees_kampioen
	IG	rusland spanje russische russen ek zweden ploeg halve griekenland kwartfinale	halve_finale andrei_arsjavin lagerb_ck roman_pavljoetsjenko luis_aragones nederlandse_bondscoach russische_ploeg otto_rehhagel lars_lagerb russische_elftal	halve_finale rusland spanje andrei_arsjavin russen roman_pavljoetsjenko lagerb_ck ek nederlandse_bondscoach russische_ploeg
	Oddsra	bondscoach ek nederland oranje ploeg rusland russen spanje	andrei_arsjavin bondscoach_marco ek_voetbal europees_kampioen europees_kampioenschap arjen_robben europese_titel halve_finale	andrei_arsjavin bondscoach_marco nederland ek_voetbal oranje europees_kampioen ploeg europees_kampioenschap

	tio	kwartfinale voetbal	nederlands_elftal roman_pavljoetsjenko	arjen_robben rusland
	Relief	rusland oranje spanje bondscoach wedstrijd russische ek trainer basten marco	bondscoach_marco arjen_robben nederlands_elftal leo_beenhakker komend_seizoen wesley_sneijder real_madrid russische_voetbalelftal khalid_boulahrouz orlando_engelaar	rusland bondscoach_marco arjen_robben oranje leo_beenhakker spanje nederlands_elftal komend_seizoen wedstrijd russische_voetbalelftal
	Ripper	rusland spanje russische ploeg arsjavin coach nederland winnaar basten rust	halve_finale andrei_arsjavin nederlands_elftal europees_kampioenschap lagerb_ck beste_speler russische_elftal titelverdediger_griekenland otto_rehhagel russische_voetbalelftal	rusland spanje halve_finale andrei_arsjavin nederlands_elftal europees_kampioenschap lagerb_ck beste_speler russische_elftal ploeg
	SVM	russische ck arsjavin rusland spanje poule zuid-korea sneijder kleine rehhagel	zwitserse_stad russische_voetballers verloren_halve russische_voetbalelftal titelverdediger_griekenland russische_spelers russische_ploeg zenit_sint russische_voetbal russische_elftal	zwitserse_stad arsjavin russische_voetballers rusland verloren_halve spanje titelverdediger_griekenland ck russische_voetbalelftal zenit_sint
Rita Verdonk	BoosTexter	binnen beveiliging trots partij man nooit hand probleem groenlinks sinke	nationaal_co mail_artikel nederland_ton terrorismebestrijding_nctb minister_ernst geert_wilders zware_persoonsbeveiliging onderzoeker_maurice hirsi_ali ge_nformeerd	nationaal_co politieke_beweging mail_artikel binnen nederland_ton terrorismebestrijding_nctb trots geert_wilders beveiliging tweede
	IG	trots politica beveiliging sinke persoonsbeveiliging beweging ton nctb dreiging brink	politieke_beweging nederland_ton nationaal_co co_rdinator rdinator_terrorismebestrijding terrorismebestrijding_nctb tweede_kamerlid den_brink minister_ernst beweging_trots	politieke_beweging nederland_ton politica nationaal_co co_rdinator beveiliging trots terrorismebestrijding_nctb sinke den_brink

	Oddsratio	beveiliging den haag minister kamer nederland partij politieke trots goed	co_rdinator den_haag ernst_hirsch geert_wilders hirsch_ballin nationaal_co nederland_ton politieke_beweging tweede_kamer tweede_kamerlid	co_rdinator beveiliging den_haag ernst_hirsch geert_wilders minister nederland_ton partij politieke_beweging tweede_kamer
	Relief	nederland den haag minister tweede kabinet kamer trots werk wilders	tweede_kamer den_haag wouter_bos geert_wilders buitenlandse_zaken verenigde_staten maxime_verhagen politieke_partijen miljoen_euro jan_marijnissen	nederland tweede_kamer den_haag wouter_bos minister geert_wilders buitenlandse_zaken verenigde_staten kabinet politieke_partijen
	Ripper	trots beveiliging ton nederland politica beweging sinke haag persoonsbeveil iging peiling	politieke_beweging nederland_ton nationaal_co co_rdinator mail_artikel hirsch_ballin miljoen_euro tweede_kamerlid zetels_halen gehouden_vanwege	trots politieke_beweging nederland_ton co_rdinator politica nationaal_co hirsch_ballin sinke beveiliging miljoen_euro
	SVM	persoonsbeveil iging brink trots politica sinke inmiddels nctb tournee adviseur rdinator	woordvoerder_kay voorzorg_binnen tienduizenden_euro's tv-programma_knevel vvd-fractievoorzitter_mark stapt_volgende persoonlijk_adviseur nina_brink politieke_beweging nederland_ton	voorzorg_binnen woordvoerder_kay persoonsbeveiliging sinke politica vvd-fractievoorzitter_mark tv-programma_knevel inmiddels tienduizenden_euro's trots
Marco van Basten	BoosTexter	bondscoach oranje voetbal nederlands nederland rusland zwitserland tweede robin europees	nederlands_elftal arjen_robben real_madrid johan_cruijff khalid_boulahrouz eerste_wedstrijd wereldkampioen_itali andr_ooijer ek_voetbal roberto_donadoni	nederlands_elftal oranje bondscoach arjen_robben real_madrid khalid_boulahrouz voetbal nederland johan_cruijff ek_voetbal
	I	bondscoach nederlands	nederlands_elftal arjen_robben	nederlands_elftal bondscoach

	G	elftal oranje itali lausanne frankrijk roemeni robben ek	khalid_boulahrouz stade_olympique eerste_wedstrijd mario_melchiot wereldkampioen_itali andr_ooijer successen_viert elftal_successen	arjen_robben khalid_boulahrouz stade_olympique oranje lausanne mario_melchiot frankrijk wereldkampioen_itali
	Oddsratio	bondscoach ek elftal frankrijk itali nederland nederlands oranje spelers wedstrijd	arjen_robben ek_voetbal europees_kampioenschap guus_hiddink khalid_boulahrouz nederlands_elftal eerste_wedstrijd mario_melchiot orlando_engelaar wereldkampioen_itali	arjen_robben bondscoach ek_voetbal europees_kampioenschap frankrijk guus_hiddink nederland nederlands_elftal eerste_wedstrijd wereldkampioen_itali
	Relief	bondscoach seizoen kreeg nooit echt kwam afgelopen hiddink ek guus	guus_hiddink leo_beenhakker komend_seizoen fc_twente europees_kampioenschap den_haag halve_finale wesley_sneijder fc_groningen miljoen_euro	guus_hiddink bondscoach leo_beenhakker komend_seizoen fc_twente europees_kampioenschap echt kreeg nooit den_haag
	Ripper	bondscoach nederlands oranje itali ruud later nederland	nederlands_elftal arjen_robben wesley_sneijder khalid_boulahrouz stade_olympique elftal_successen wereldkampioen_itali eerste_wedstrijd guus_hiddink europese_titel	bondscoach nederlands_elftal oranje arjen_robben wesley_sneijder khalid_boulahrouz stade_olympique elftal_successen wereldkampioen_itali eerste_wedstrijd
	SVM	bondscoach nederlands hiddink lausanne oostenrijk stelde wereldkampio en andr melchiot orlando	vervanger_oproepen stade_olympique uur_spelen spelers_rust tien_dagen rond_uur victor_piturca laatste_training nederland_wint rinus_michels	bondscoach vervanger_oproepen uur_spelen stade_olympique tien_dagen nederlands hiddink spelers_rust lausanne stelde
Geert Wilders	BoosT	pvv leven fitna privacy goed heisa nederlandse	film_fitna politiek_privacy rita_verdonk tweede_kamerlid voorpagina_binnenland nederland_amerikaanse openbaar_ministerie	pvv film_fitna politiek_privacy leven rita_verdonk tweede_kamerlid voorpagina_binnenland

	exter	kamer vrijheid tweede	nederlandse_politici politieke_partijen amerikaanse_presidentenbiografie	goed nederland_amerikaanse heisa
	IG	film privacy gematigde moslimextremisten presidentenbiografie oud-premierswie oud-politiciwelke koranfilmwaar oud-premiers buitenlandwelke	nederland_amerikaanse mysterie_hollowaylees nederlanderswelke_nederlanders tweede-kamerledenwelke_tweede-kamerleden kredietcrisis_race nederlandse_politici klik_zoek lokaal_bestuurwie kredietcrisisnieuws_achtergronden land_invloedrijke	nederland_amerikaanse film mysterie_hollowaylees privacy gematigde nederlanderswelke_nederlanders tweede-kamerledenwelke_tweede-kamerleden moslimextremisten nederlandse_politici kredietcrisis_race
	Oddsratio	film kabinet land nederland artikel nederlandse politieke privacy pvda volgende	amerikaanse_presidentenbiografie brusselwelke_nederlandse buitenland_vertrokken defensietopwelke_mannen kredietcrisis_race mysterie_hollowaylees nederland_amerikaanse nederlandse_politici politieke_partijen bestuurwelke_politici	amerikaanse_presidentenbiografie film brusselwelke_nederlandse kabinet buitenland_vertrokken land defensietopwelke_mannen artikel nederland_amerikaanse politieke_partijen
	Relief	minister nederland kabinet kamer tweede film haag zaken den balkenende	wouter_bos nederland_amerikaanse nederlanderswelke_nederlanders tweede-kamerledenwelke_tweede-kamerleden nederlandse_politici mysterie_hollowaylees miljoen_euro politieke_partijen tweede-kamerleden_domineren kredietcrisisnieuws_achtergronden	minister wouter_bos kabinet kamer nederland_amerikaanse tweede film nederlandse_politici mysterie_hollowaylees miljoen_euro
	Ripper	film fitna gematigde kinderporno privacy pvv-leider lutser pvv partij oorlog	nederlandse_politici film_fitna rita_verdonk politie_politiek peak_oil politiek_privacy ernst_hirsch nederland_amerikaanse amerikaanse_presidentenbiografie embryoselectie_europa	gematigde nederlandse_politici film_fitna rita_verdonk kinderporno ernst_hirsch politie_politiek pvv-leider peak_oil politiek_privacy
	SV	fitna pvv-leider zoekterm	film_fitna politiek_privacy dood_downloaden	pvv-leider film_fitna politiek_privacy

	M	abonneer onderschrift tweede-kamerledenwelke nieuwsbrief veo peak-oil gematigde	pvda_reflectomaus nederlandse_politici geloof_god heerst_hollandse china_christenunie kinderporno_koppenklopper usa_veo	zoekterm dood_downloaden abonneer onderschrift heerst_hollandse nederlandse_politici pvda_reflectomaus

Table A1: Top 10 words

B Computational time

This appendix provides the exact time required to produce the 10 words for each entity-technique for a cross validation fold. We used stratified 5-fold cross validation. For each fold the time to produce the 10 words can be different, that's why took the minimum and maximum number of the 5-fold cross validation. Table A2 provides the minimum time and the maximum time that was required for a cross validation fold for each entity-technique for both single and two consecutive words.

Entity	Technique	Time	
		Single words	Two consecutive words
PH	BoosTexter	8 seconds – 10 seconds	6 seconds – 7 seconds
	IG	0 seconds – 1 second	0 seconds – 1 second
	Oddsratio	0 seconds	0 seconds
	Relief	0 seconds – 1 second	1 second
	Ripper	1 second – 2 seconds	1 second
	SVM	4 seconds – 6 seconds	1 second – 3 seconds
SD	BoosTexter	13 seconds – 10 seconds	8 seconds – 9 seconds
	IG	0 seconds – 1 second	0 seconds – 1 second
	Oddsratio	0 seconds	0 seconds
	Relief	1 second – 2 seconds	1 second – 2 seconds
	Ripper	1 second – 2 seconds	1 second – 2 seconds
	SVM	25 seconds – 45 seconds	3 seconds – 5 seconds
BS	BoosTexter	17 seconds – 19 seconds	14 seconds – 16 seconds
	IG	0 seconds – 1 second	0 seconds – 1 second
	Oddsratio	0 seconds	0 seconds
	Relief	2 seconds – 3 seconds	2 seconds – 3 seconds
	Ripper	2 seconds – 3 seconds	2 seconds – 3 seconds
	SVM	21 seconds – 46 seconds	5 seconds – 10 seconds
AA	BoosTexter	12 seconds – 13 seconds	11 seconds – 13 seconds
	IG	0 seconds – 1 second	0 seconds – 1 second
	Oddsratio	0 seconds	0 seconds
	Relief	7 seconds – 8 seconds	7 seconds – 8 seconds
	Ripper	2 seconds – 3 seconds	2 seconds – 4 seconds
	SVM	15 seconds – 22 seconds	5 seconds – 9 seconds
M	BoosTexter	20 seconds – 21 seconds	17 seconds – 18 seconds
	IG	0 seconds – 1 second	0 seconds – 1 second
	Oddsratio	0 seconds	0 seconds
	Relief	5 seconds – 6 seconds	4 seconds – 5 seconds
	Ripper	3 seconds – 5 seconds	3 seconds – 5 seconds
	SVM	1 minute and 43 second – 2 minutes and 50 seconds	10 seconds – 23 seconds
EvdS	BoosTexter	27 seconds – 34 seconds	22 seconds – 24 seconds
	IG	1 second	0 seconds – 1 second

	Oddsratio	0 seconds	0 seconds
	Relief	42 seconds – 43 seconds	41 seconds – 42 seconds
	Ripper	14 seconds – 18 seconds	15 seconds – 21 seconds
	SVM	10 minutes and 7 seconds – 27 minutes and 13 seconds	1 minute and 39 seconds – 3 minutes and 43 second
AK	BoosTexter	31 seconds – 35 seconds	24 seconds – 25 seconds
	IG	1 second	1 second – 3 seconds
	Oddsratio	0 seconds	0 seconds
	Relief	52 seconds – 54 seconds	52 seconds – 54 seconds
	Ripper	5 seconds – 6 seconds	17 seconds – 22 seconds
	SVM	1 minute and 33 seconds – 2 minutes and 11 seconds	1 minute and 5 seconds – 2 minutes and 37 seconds
WS	BoosTexter	47 seconds – 1 minute and 38 seconds	36 seconds – 38 seconds
	IG	1 second – 2 seconds	1 second – 2 seconds
	Oddsratio	0 seconds – 1 second	0 seconds
	Relief	1 minute and 50 seconds – 1 minute and 53 seconds	1 minute and 46 seconds – 1 minute and 49 seconds
	Ripper	20 seconds – 29 seconds	20 seconds – 29 seconds
	SVM	44 minutes and 4 seconds – 1 hour and 19 minutes	15 minutes and 14 seconds – 33 minutes and 17 seconds
GH	BoosTexter	56 seconds – 59 seconds	41 seconds – 43 seconds
	IG	1 second – 2 seconds	1 second – 2 seconds
	Oddsratio	0 seconds	0 seconds
	Relief	3 minutes and 9 seconds – 3 minutes and 12 seconds	3 minutes and 5 seconds – 3 minutes and 11 seconds
	Ripper	30 seconds – 34 seconds	52 seconds – 1 minute and 3 seconds
	SVM	26 minutes and 54 seconds – 44 minutes and 38 seconds	11 minutes and 10 seconds – 22 minutes and 46 seconds
RV	BoosTexter	1 minute and 9 seconds – 1 minute and 15 seconds	49 seconds – 52 seconds
	IG	1 second – 2 seconds	1 second – 2 seconds
	Oddsratio	0 seconds	0 seconds – 1 second
	Relief	4 minutes and 13 seconds – 4 minutes and 24 seconds	4 minutes and 7 seconds – 4 minutes and 14 seconds
	Ripper	41 seconds – 54 seconds	1 minute and 5 seconds – 1 minute and 29 seconds
	SVM	1 hour and 2 minutes – 1 hour and 55 minutes	45 minutes and 48 seconds – 3 hours and 21 minutes
MvB	BoosTexter	1 minute and 26 seconds – 1 minute and 30 seconds	1 minute and 1 second – 1 minute and 3 seconds

	IG	1 second – 2 seconds	1 second – 3 seconds
	Oddsratio	0 seconds	0 seconds
	Relief	6 minutes and 21 seconds – 6 minutes and 31 seconds	6 minutes and 22 seconds – 6 minutes and 28 seconds
	Ripper	39 seconds – 46 seconds	44 seconds – 1 minute and 3 seconds
	SVM	2 hours and 21 minutes – 4 hours and 20 minutes	46 minutes and 41 seconds – 1 hour and 41 minutes
GW	BoosTexter	1 minute and 48 seconds – 1 minute and 53 seconds	1 minute and 22 seconds – 1 minute and 28 seconds
	IG	2 seconds – 4 seconds	1 second – 3 seconds
	Oddsratio	0 seconds – 1 second	0 seconds
	Relief	10 minutes and 38 seconds – 10 minutes and 46 seconds	10 minutes and 33 seconds – 10 minutes and 44 seconds
	Ripper	1 minute and 9 seconds – 1 minute and 35 seconds	25 seconds – 36 seconds
	SVM	1 hour and 19 minutes – 4 hours and 29 minutes	37 minutes and 35 seconds – 5 hours and 5 minutes

Table A2: Time requires for a cross validation fold

C Kendall's correlation coefficient

This appendix provides a description of how Kendall's correlation coeffient is calculated. Kendall's tau is computed as follows:

$$\tau = \frac{n_c - n_d}{n(n-1)/2}$$

where
n_c is the number of concordant pairs
n_d is the number of disconcordant pairs, and also equal to $[n(n-1)/2] - n_c$
n is the number of all pairs possible

The calculation of Kendall's tau will be illustrated with an example. Suppose we have the data as provided in Table A3.

	A	B	C	D	E	F	G	H	I	J
Data X	1	2	3	4	5	6	7	8	9	10
Data Y	5	6	3	2	8	1	4	9	10	7

Table A3: Example data

In Table A4 an "x" is provided if the pairs are disconcordant, while a "1" is given for pairs that are concordant.

	A	B	C	D	E	F	G	H	I	J	n_c	n_d
A	-	1	x	x	1	x	x	1	1	1	5	4
B	-	-	x	x	1	x	x	1	1	1	4	4
C	-	-	-	x	1	x	1	1	1	1	5	2
D	-	-	-	-	1	x	1	1	1	1	5	1
E	-	-	-	-	-	x	x	1	1	x	2	3
F	-	-	-	-	-	-	1	1	1	1	4	0
G	-	-	-	-	-	-	-	1	1	1	3	0
H	-	-	-	-	-	-	-	-	1	x	1	1
I	-	-	-	-	-	-	-	-	-	x	0	1
J	-	-	-	-	-	-	-	-	-	-		
SUM											29	16

Table A4: Calculating the number of (dis) concordant pairs

According to our example the Kendall's correlation coefficient

$$\tau = \frac{n_c - n_d}{n(n-1)/2} = \frac{29-16}{10(10-1)/2} = 0.29$$

D ANOVA

This appendix is taken from the Lecture Notes **[42]**.

The basic idea behind the analysis of variance (ANOVA) method is that is a statistical technique that investigates how the response variables depend on the explanatory variables. For a One-factor Model it investigates whether there exists a difference between all levels. For a Multi-factor Model (a model with two or more variables) it investigates whether these variables should be included in the model. It takes into account the size of the dataset, the degrees of freedom (Df), the residual sum of squares, and the mean sum of squares. Given this the F- test statistic is calculated. For large value of this statistic the null-hypothesis is rejected. For a One-factor Model the formulas will be given.

The general One-factor model is given by:

$$\Omega : \begin{cases} Y_{ij} = \eta_i + e_{ij} \\ Ee_{ij} = 0 \\ Cov(e_{ij,} e_{kl}) = \begin{cases} \sigma^2, & (i,j) = (k,l) \\ 0, & (i,j) \neq (k,l) \end{cases} \end{cases} \qquad \textbf{(A D-1)}$$

for $i = 0,...I, j = 1,..., J_i$

For a linear model $\eta_i = \mu + \alpha_i$, where μ is an unknown general mean and α_i is an unknown effect due to the factor having level i.

In order to uniquely determine $\beta = (u, \alpha_1,...., \alpha_I)^T$ we need to specify some constraints (see Section 3.1). We can set $\sum_{i=1}^{I} \alpha_i = 0$ or $\mu = 0$.

We will determine β by using the first constraint $\sum_{i=1}^{I} \alpha_i = 0$.

The sum of squares $S(\beta)$ is given by the following equations:

$$S(\beta) = \sum_{i=1}^{I} \sum_{j=1}^{J_i} (Y_{ij} - EY_{ij})^2 = \sum_{i=1}^{I} \sum_{j=1}^{J_i} (Y_{ij} - \mu - \alpha_i)^2 \qquad \textbf{(A D-2)}$$

Differentiating $S(\beta)$ with respect to β results in the following normal equations:

$$\begin{aligned} \frac{\partial}{\partial \mu} S(\beta) &= -2 \sum_{i=1}^{I} \sum_{j=1}^{J_i} (Y_{ij} - \mu - \alpha_i)^2 = 0 \\ \frac{\partial}{\partial \alpha_i} S(\beta) &= -2 \sum_{j=1}^{J_i} (Y_{ij} - \mu - \alpha_i)^2 = 0 \end{aligned} \qquad \textbf{(A D-3)}$$

for $i = 0,\ldots I$

We can solve **(A D-3)** by making use of the following constraint $\sum_{i=1}^{I} \alpha_i = 0$. In this case the least squares estimators for μ and α_i is then given by:

$$\hat{\mu} = \frac{1}{I}\sum_{i=1}^{I}\frac{1}{J_i}\sum_{j=1}^{J_i} Y_{ij} = Y_{..}$$
$$\hat{\alpha}_i = \frac{1}{J_i}\sum_{j=1}^{J_i} Y_{ij} - \frac{1}{I}\sum_{i=1}^{I}\frac{1}{J_i}\sum_{j=1}^{J_i} Y_{ij} = Y_{i.} - Y_{..} \qquad \textbf{(A D-4)}$$

The structure of determining β for a Multi-factor Model is the same as the one explained for the One-factor Model.

If we consider the model given in **(A D-1)** with intercept equal to zero, then we would like to know whether all levels i have the same expectation. This leads to the following hypothesis:

$H_0 : \alpha_1 = \ldots = \alpha_I$ or equivalently H_0 : the smaller model ω holds, all levels have the same expectation

$$\omega : \begin{cases} Y_{ij} = \alpha + e_{ij} \\ Ee_{ij} = 0 \\ Cov(e_{ij,} e_{kl}) = \begin{cases} \sigma^2, & (i,j) = (k,l) \\ 0, & (i,j) \neq (k,l) \end{cases} \end{cases} \qquad \textbf{(A D-5)}$$

We can calculate for both models Ω and ω the sum of squares $S(\beta)$. This can be summarized in ANOVA table:

Sum of Squares	Df	Mean Sum of Squares	F
$S_\omega - S_\Omega$	$I-1$	$\frac{S_\omega - S_\Omega}{I-1}$	$\frac{(S_\omega - S_\Omega)/(I-1)}{S_\Omega/(n-I)}$
S_Ω	$n-I$	$\frac{S_\Omega}{n-I}$	

S_Ω is the sum of squares within groups
S_ω is the sum of squares of total variation around the general mean

S_ω - S_Ω is the sum of squares between groups

The test statistic F under H_0 is given as follows:

$$F = \frac{(S_\omega - S_\Omega)/(I-1)}{S_\Omega/(n-I)} \sim F_{I-1,n-I}$$

The null-hypothesis is rejected for large values of F.

How will ANOVA be applied will be explained with an example. Suppose we want to know whether there is significant difference between different techniques when looking at the (average) scores provided by humans. This can be done by using the One-factor ANOVA Model. The number of levels is in this case equal to the number of techniques used. If we assume that the data would look like the one provided in Table A5.

Entity	Average score by humans	Technique
PH	7.8	BoosTexter
PH	6	IG
PH	7	Ripper
PH	3	Relief
PH	5	SVM
SD	8	BoosTexter
SD	7.5	IG
SD	3.5	Ripper
SD	6	Relief
SD	8	SVM
BS	5.5	BoosTexter
etc.		

Table A5: Part of the data

The response variable would here be the average score and the explanatory variable would be the technique.

Note that ANOVA can also be applied to other evaluation measures such as the F_1-measure. The ANOVA function in R will be used.

E Code

The code for calling BoosTexter, the feature selection techniques in Weka, and the SVM-class in Weka is provided in this Appendix. Also the code that is used in R.

Code for calling BoosTexter:
```
Boostexter_train.exe <dir> <number of iterations>
```

Where dir is the directory where the following files:
- class_train.txt contains document ids and their corresponding class
- voc.txt contains word ids with their corresponding words
- freqMatr_train.txt contains only 0's and 1's. Vertically the document id's are given and horizontally the word ids.
- number of iterations (In Figure 10 as T) is 100

are and also where the output will be stored.

The top 10 words are selected with the highest weight from all the 100 iterations.

Code for calling IG:
```
java  -Xmx1512m -classpath D:\Users\Priya\WEKA\Weka-3-4\weka.jar
weka.attributeSelection.InfoGainAttributeEval -s
"weka.attributeSelection.Ranker -T 0.0 -N 10 "  -i
data_train.arff
```

Code for calling JRip:
```
java  -Xmx1512m -classpath D:\Users\Priya\WEKA\Weka-3-4\weka.jar
weka.classifiers.rules.JRip  -t data_train.arff
```

Code for calling Relief:
```
java  -Xmx1512m -classpath D:\Users\Priya\WEKA\Weka-3-4\weka.jar
weka.attributeSelection.ReliefFAttributeEval -s
"weka.attributeSelection.Ranker -T 0.0 -N 10 "  -i
data_train.arff
```

Code for calling SVM-FS:
```
java  -Xmx1512m -classpath D:\Users\Priya\WEKA\Weka-3-4\weka.jar
weka.attributeSelection.SVMAttributeEval -X 10 -Y 0 -Z 0 -P 1.0E-
25 -T 1.0E-10 -C 1.0 -N 0 -s "weka.attributeSelection.Ranker -T
0.0 -N 10 "  -i data_train.arff
```

Code for calling SVM-Class:
```
java  -Xmx1512m -classpath D:\Users\Priya\WEKA\Weka-3-4\weka.jar
weka.classifiers.functions.SMO -t   train.arff  -T  test.arff
```

Note that the train and test arff files are the files containing only the selected words as attributes.

The results are analyzed in **R**. The R code is given below:

###################### Nominal concordance ######################

```
nomconcdata <- read.table("D:\\Users\\Priya\\R-
2.7.1\\data\\nom_conc.txt ", header = TRUE, sep = "\t")

par(mfrow=c(3,1))

plot(Nom_conc_SW~ Technique, data = nomconcdata, ylim=c(0,1),
xlab="feature selection technique", ylab="nominal concordance")
title("Box-plot of nominal concordance of 5 entities for single
words",  cex.main =1.2, font.main = 4, col.main= "blue")

plot(Nom_conc_TCW~ Technique, data = nomconcdata,  ylim=c(0,1),
xlab="feature selection technique", ylab="nominal concordance")
title("Box-plot of nominal concordance of 5 entities for two
consecutive words",  cex.main =1.2, font.main = 4, col.main=
"blue")

plot(Nom_conc_CW~ Technique, data = nomconcdata,  ylim=c(0,1),
xlab="feature selection technique", ylab="nominal concordance")
title("Box-plot of nominal concordance of 5 entities for composed
words",  cex.main =1.2, font.main = 4, col.main= "blue")

nomconcdata <- read.table("D:\\Users\\Priya\\R-
2.7.1\\data\\nom_conc_withoutSVM.txt", header = TRUE, sep = "\t")

data.aov<- aov(Nom_conc_SW ~ Technique, data = nomconcdata)
summary(data.aov)
data.aov<- aov(Nom_conc_TCW~ Technique, data = nomconcdata)
summary(data.aov)
data.aov<- aov(Nom_conc_CW~ Technique, data = nomconcdata)
summary(data.aov)
```

################################### F_1 – measure ################################

```
par(mfrow=c(3,1))

f1measuredata <- read.table("D:\\Users\\Priya\\R-
2.7.1\\data\\f1measure.txt", header = TRUE, sep = "\t")
plot(F1_measure_SW ~ Technique, data = f1measuredata,
xlab="feature selection technique", ylab="F1-measure",
ylim=c(0,1))
title("Box-plot of F1-measure of all 12 entities for single
words",  cex.main =1.2, font.main = 4, col.main= "blue")

plot( F1_measure_TCW ~  Technique, data = f1measuredata,
xlab="feature selection technique", ylab="F1-measure",
ylim=c(0,1))
```

```
title("Box-plot of F1-measure of all 12 entities for two
consecutive words",  cex.main =1.2, font.main = 4, col.main=
"blue")

plot(F1_measure_CW ~ Technique, data = f1measuredata,
xlab="feature selection technique", ylab="F1-measure",
ylim=c(0,1))
title(" Box-plot of F1-measure of all 12 entities for composed
word lists",  cex.main =1.2, font.main = 4, col.main= "blue")

data.aov<- aov(F1_measure_SW ~ Technique, data = f1measuredata)
summary(data.aov)
data.aov<- aov(F1_measure_TCW~ Technique, data = f1measuredata)
summary(data.aov)
data.aov<- aov(F1_measure_CW~ Technique, data = f1measuredata)
summary(data.aov)


###################### Absolute difference between correlations##################

par(mfrow=c(3,1))

ACDdata <- read.table("D:\\Users\\Priya\\R-
2.7.1\\data\\correlation_diff.txt", header = TRUE, sep = "\t")
plot(ACD_SW ~ Technique, data = ACDdata,  xlab="feature selection
technique", ylab="Absolute difference", ylim=c(0,1))
title("Absolute difference in correlation of all 12 entities for
single words",  cex.main =1.2, font.main = 4, col.main= "blue")

plot(ACD_TCW ~  Technique, data = ACDdata,  xlab="feature
selection technique", ylab="Absolute difference", ylim=c(0,1))
title("Absolute difference in correlation of all 12 entities for
two consecutive words",  cex.main =1.2, font.main = 4, col.main=
"blue")

plot(ACD_CW ~ Technique, data = ACDdata,  xlab="feature selection
technique", ylab="Absolute difference", ylim=c(0,1))
title("Absolute difference in correlation of all 12 entities for
composed word lists",  cex.main =1.2, font.main = 4, col.main=
"blue")


data.aov<- aov(ACD_SW ~ Technique, data = ACDdata)
summary(data.aov)
data.aov<- aov(ACD_TCW ~ Technique, data = ACDdata)
summary(data.aov)
data.aov<- aov(ACD_CW ~ Technique, data = ACDdata)
summary(data.aov)
```

###################### Computation of the correlations #######################

```
#Calulating the correlation for Spearman

calculate_corr_all_scores <- function (data, name) {
      output<-c()
      number_significant =0;
      total = 0;
      for(i in 2:18) {
            for(j in (i+1):19) {
                  x = cor(data[,i],data[,j], method ="spearman");
                  output <- c(output, x);
                  significance =rcorr(data[,i],data[,j],
                  type="spearman")$P[1,2]
                  if(significance <= 0.05){
                              number_significant =
                              number_significant + 1;
                  }
                  total = total +1;
            }
      }
      result <-c()
      result[name] = mean(output)
      final_output <- list(result, number_significant,
(number_significant/total))
}

# Example Paris Hilton
# Note that the rest of the entities go in the same way
data <- read.table("D:\\Users\\Priya\\Finalrun\\Scores\\PH.txt",
header=TRUE,sep ="\t")
x = calculate_corr_all_scores (data, "PH")

sink(file="D:\\Users\\Priya\\Finalrun\\Scores\\Spearman_Correlati
on_scores.txt")
"PH"
"Average correlation"
x[[1]]
"Number significant"
x[[2]]
"Ratio significant"
x[[3]]


# Calulating Kendall’s correlation coefficient

calculate_corr_all_scores <- function (data, whichmethod, name) {
      output<-c()
      number_significant =0;
      total = 0;
      for(i in 2:18) {
```

```
            for(j in (i+1):19) {
                  x = Kendall(data[,i], data[,j])$tau[1]
                  output <- c(output, x);
                  significance =Kendall(data[,i], data[,j])$sl[1]
                  if(significance <= 0.05){
                        number_significant = number_significant +
1;
                  }
                  total = total +1;
            }
      }
      result <-c()
      result[name] = mean(output)
      final_output <- list(result, number_significant,
(number_significant/total))
}

# Example Paris Hilton
# Note that the rest of the entities go in the same way
data <- read.table("D:\\Users\\Priya\\Finalrun\\Scores\\PH.txt",
header=TRUE,sep ="\t")
x = calculate_corr_all_scores (data, "PH")

sink(file="D:\\Users\\Priya\\Finalrun\\Scores\\Kendall_Correlatio
n_scores.txt")
"PH"
"Average correlation"
x[[1]]
"Number significant"
x[[2]]
"Ratio significant"
x[[3]]
```

########################## Example Part of the data file #####################

Words	Tim	Stijn	Paul	Mark	Ineke	Menno	Coen	Hans	Vicky	Renuka	Peter
	Bilal	Gabriel	Andjalie	Arun	Marten	Mathijs	Alicia				
zwanger	2	2	1	2	1	2	2	0	0	2	2
	2	0	1	0	0	2	0				
vriendje_benji		0	2	0	0	0	0	0	0	0	0
	0	0	0	0	2	2	1	0			
vriendje	2	0	0	0	1	2	0	0	0	1	2
	0	0	1	0	1	0	0				
vriend_steven		0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0			

F Results Spearman correlation coefficient

The results of applying Spearman correlation coefficient instead of Kendall's correlation coefficient can be found in Table A66, Figure A1, and Table A7.

Entity	Technique	Correlation Spearman			P-values		
		SW	TCW	CW	SW	TCW	CW
PH	BoosTexter	0.15	0.25	0.16	0.373	0.205	0.205
	IG	0.10	0.12	0.02	0.558	0.559	0.903
	Oddsratio	-0.17	-0.08	0.04	0.329	0.674	0.783
	Relief	0.55	0.37	0.27	0.001	0.059	0.029
	Ripper	0.06	0.34	0.08	0.708	0.083	0.516
	SVM	-0.01	0.14	0.01	0.938	0.477	0.947
SD	BoosTexter	0.27	-0.05	0.18	0.096	0.763	0.115
	IG	-0.04	0.42	0.04	0.813	0.007	0.733
	Oddsratio	0.12	0.07	0.13	0.460	0.662	0.254
	Relief	0.07	0.11	0.24	0.686	0.511	0.031
	Ripper	0.33	0.08	0.08	0.044	0.635	0.496
	SVM	0.01	-0.22	-0.19	0.973	0.176	0.092
BS	BoosTexter	-0.21	0.51	0.46	0.300	0.011	0.001
	IG	0.04	0.53	0.31	0.837	0.008	0.028
	Oddsratio	-0.19	0.40	0.16	0.354	0.052	0.257
	Relief	0.01	0.54	0.28	0.974	0.006	0.050
	Ripper	-0.04	0.35	0.43	0.843	0.091	0.002
	SVM	-0.22	0.05	0.11	0.270	0.828	0.468
AA	BoosTexter	0.12	0.17	0.28	0.487	0.365	0.024
	IG	0.30	0.30	0.22	0.085	0.111	0.080
	Oddsratio	-0.12	0.26	0.19	0.502	0.168	0.130
	Relief	0.14	0.27	0.21	0.434	0.155	0.106
	Ripper	0.45	0.22	0.20	0.008	0.259	0.111
	SVM	0.26	0.24	0.17	0.132	0.219	0.189
M	BoosTexter	-0.21	-0.02	-0.18	0.237	0.917	0.148
	IG	0.05	0.42	0.19	0.787	0.014	0.122
	Oddsratio	0.10	0.27	0.13	0.550	0.119	0.303
	Relief	0.11	0.12	0.21	0.520	0.488	0.087
	Ripper	0.09	-0.15	0.11	0.595	0.413	0.365
	SVM	0.00	-0.02	0.22	0.989	0.924	0.071
EvdS	BoosTexter	-0.01	0.01	0.25	0.974	0.960	0.030
	IG	0.15	0.35	0.31	0.355	0.037	0.008
	Oddsratio	0.16	-0.08	-0.03	0.341	0.667	0.802
	Relief	0.18	0.16	0.24	0.278	0.348	0.039
	Ripper	0.35	0.19	0.20	0.029	0.279	0.083
	SVM	0.18	-0.22	0.03	0.285	0.208	0.779
AK	BoosTexter	-0.06	0.29	0.16	0.731	0.100	0.179

	IG	0.42	-0.10	0.25	0.007	0.580	0.032
	Oddsratio	0.03	0.24	0.17	0.866	0.186	0.162
	Relief	0.30	0.31	0.30	0.062	0.082	0.009
	Ripper	0.00	0.26	0.26	0.984	0.150	0.025
	SVM	0.48	0.34	0.33	0.002	0.054	0.005
WS	BoosTexter	-0.21	-0.12	-0.14	0.237	0.479	0.240
	IG	-0.08	0.22	0.20	0.667	0.196	0.097
	Oddsratio	0.25	0.11	0.09	0.155	0.516	0.467
	Relief	0.24	0.38	0.26	0.161	0.024	0.028
	Ripper	0.36	0.09	0.23	0.035	0.602	0.053
	SVM	-0.28	0.09	0.05	0.102	0.604	0.690
GH	BoosTexter	0.24	0.23	0.30	0.207	0.189	0.016
	IG	0.01	0.16	0.25	0.969	0.369	0.044
	Oddsratio	0.43	0.23	0.17	0.018	0.193	0.169
	Relief	0.23	-0.16	0.11	0.229	0.358	0.388
	Ripper	0.07	0.36	0.30	0.715	0.038	0.014
	SVM	-0.18	0.20	-0.01	0.353	0.265	0.909
RV	BoosTexter	-0.12	-0.11	0.00	0.508	0.513	0.981
	IG	0.47	-0.03	0.12	0.005	0.877	0.337
	Oddsratio	0.06	0.15	0.20	0.732	0.386	0.092
	Relief	0.12	0.31	0.34	0.494	0.059	0.004
	Ripper	0.56	0.01	0.15	0.001	0.958	0.210
	SVM	0.24	-0.06	0.12	0.171	0.714	0.332
MvB	BoosTexter	0.40	0.20	0.40	0.027	0.278	0.001
	IG	0.13	-0.10	-0.05	0.498	0.578	0.720
	Oddsratio	0.43	0.31	0.26	0.015	0.091	0.041
	Relief	-0.22	0.07	-0.07	0.241	0.690	0.576
	Ripper	0.15	0.10	0.06	0.430	0.589	0.638
	SVM	-0.18	-0.17	-0.19	0.334	0.373	0.139
GW	BoosTexter	0.40	0.60	0.30	0.008	0.000	0.008
	IG	-0.01	-0.17	0.05	0.940	0.311	0.649
	Oddsratio	0.24	-0.16	0.01	0.123	0.346	0.964
	Relief	0.10	0.06	0.15	0.521	0.713	0.196
	Ripper	0.22	0.18	0.13	0.154	0.303	0.244
	SVM	-0.07	0.24	0.18	0.668	0.160	0.104

Table A6: Spearman correlation coefficient for 12 entities

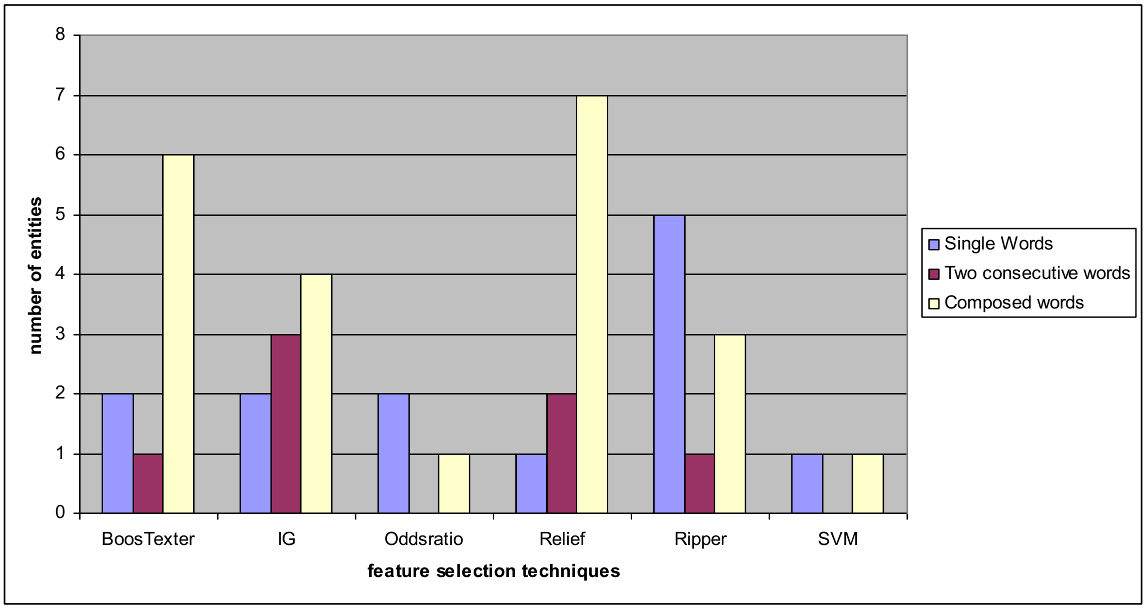

Figure A1: Number of times a significant positive correlation was found between a technique and humans

Entity	Spearman correlation	
	Correlation coefficient	Ratio significant
PH	0.36	0.73
SD	0.47	0.93
BS	0.33	0.56
AA	0.26	0.54
M	0.36	0.80
EvdS	0.46	0.87
AK	0.28	0.56
WS	0.38	0.82
GH	0.25	0.51
RV	0.19	0.39
MvB	0.32	0.65
GW	0.40	0.77

Table A7: Average Spearman correlation coefficient for 18 persons

Comparing the results of Table A66, Figure A1, and Table A7 with the ones in Table 16, Figure 18: Figure 18 and Table 14 respectively, we see that there is almost no difference between using the Kendall's correlation coefficient (and test) and the Spearman correlation coefficient (and test).

G Total human scores

In this Appendix the frequency tables are provided for each entity. Note that only those words that had a score higher than zero are taken.

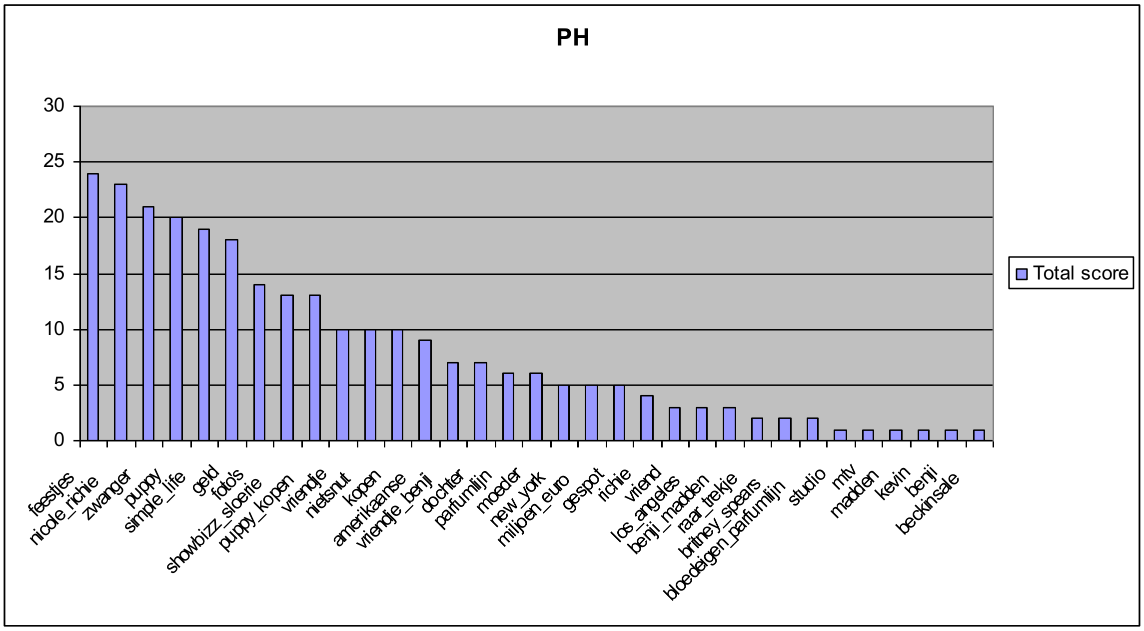

Figure A2: Scores for PH

It is obvious from Figure A2 that the word “nicole_ritchie” (score of 23) is preferred above “Ritchie” (score of 4). For the word “puppy” (score of 20) this preference is also obvious, because the word “puppy_kopen” have a score of 13 and the word “kopen” have a score of 10. It is clear that the word “vriendje_benji” (score of 7) is more favored than the word “benji” (score of 1) itself, but that the word “vriendje” (score of 10) and “vriendje_benji” are almost equally preffered.

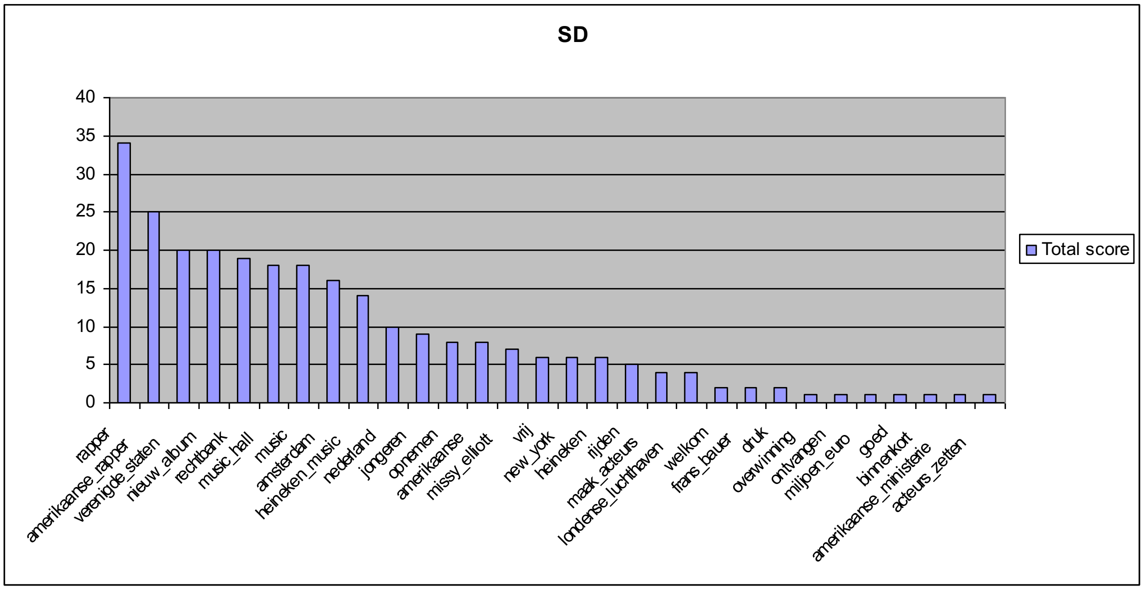

Figure A3: Scores for SD

It is not so obvious from Figure A3 that the word “rapper” (score of 34) is preferred above “amerikaanse_rapper” (score of 25), because this “amerikaanse_rapper” is the second word that is most favored in the total list. It is clear that the word “heineken_music” (score of 14) is more desired than the word “heineken” (score of 6) itself.

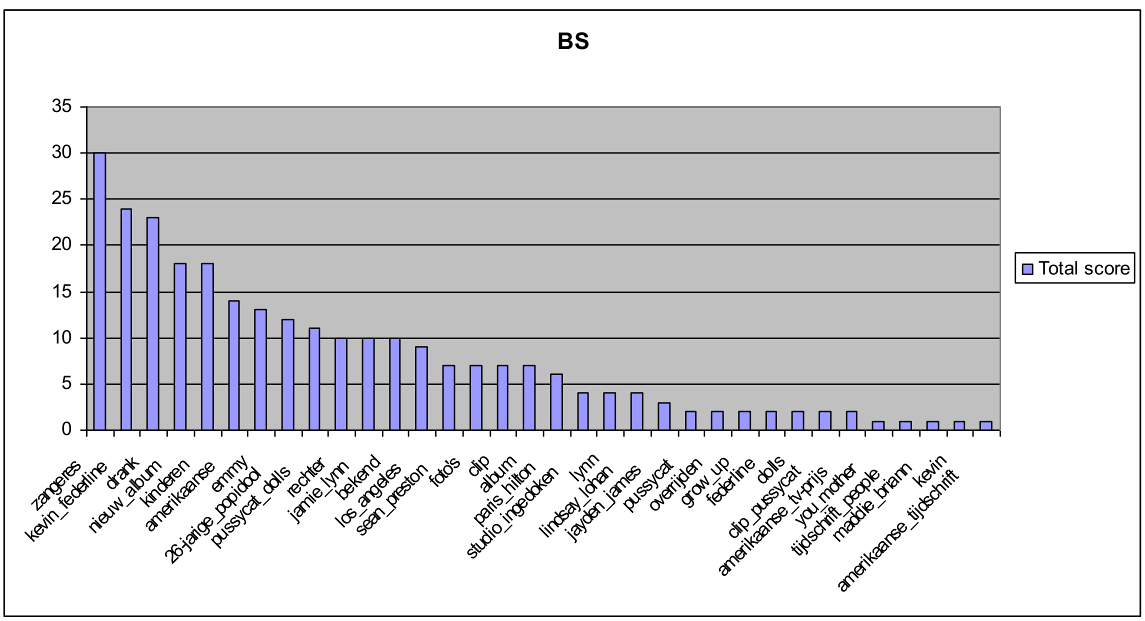

Figure A4: Scores for BS

It is so obvious from Figure A4 that the word “pussycat_dolls” (score of 11) is preferred above the two words “pussycat” (score of 2) and “dolls” (score of 2). It is clear that the word “kevin_federline” (score of 24) is more favored than the words “federline” (score of 2) and “Kevin” (score of 1) itself.

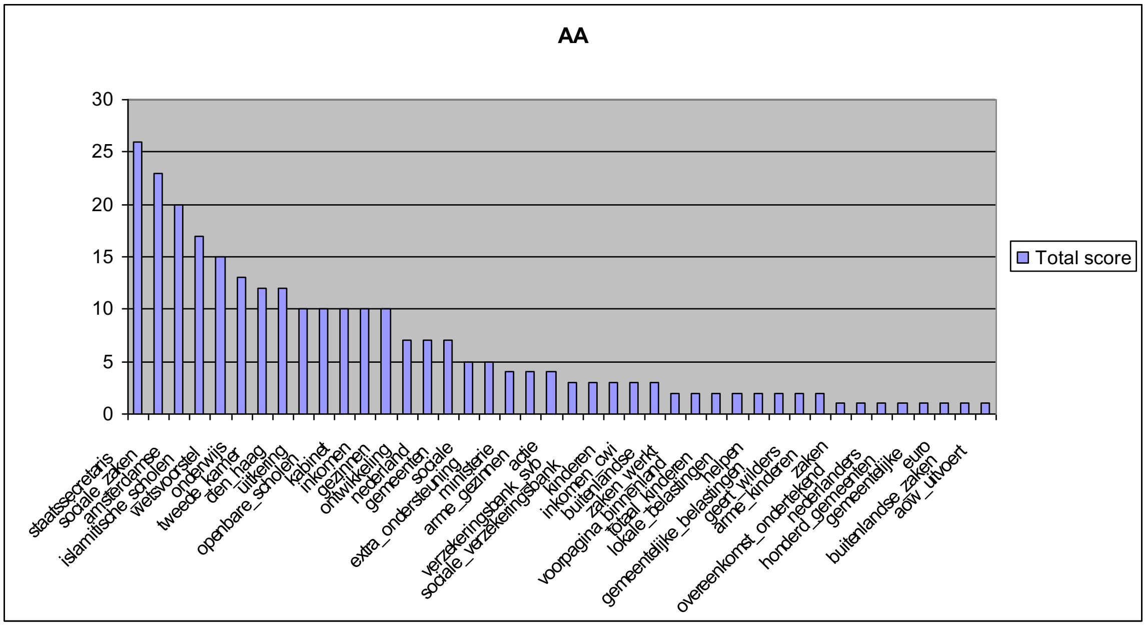

Figure A5: Scores for AA

It is obvious from Figure A5 that the word “gezinnen” (score of 10) is preferred above the word “arme_gezinnen” (score of 4). It is also clear that the word “inkomen” (score of 10) is more favored than the word “inkomen_cwi” (score of 3) itself. Another word that is more preferred is “sociale_zaken” (score of 23) above “zaken” (score of 1) and “sociale” (score of 5). One more word that is desired is “gemeenten” (score of 7) above “honderd_gemeenten” (score of 1).

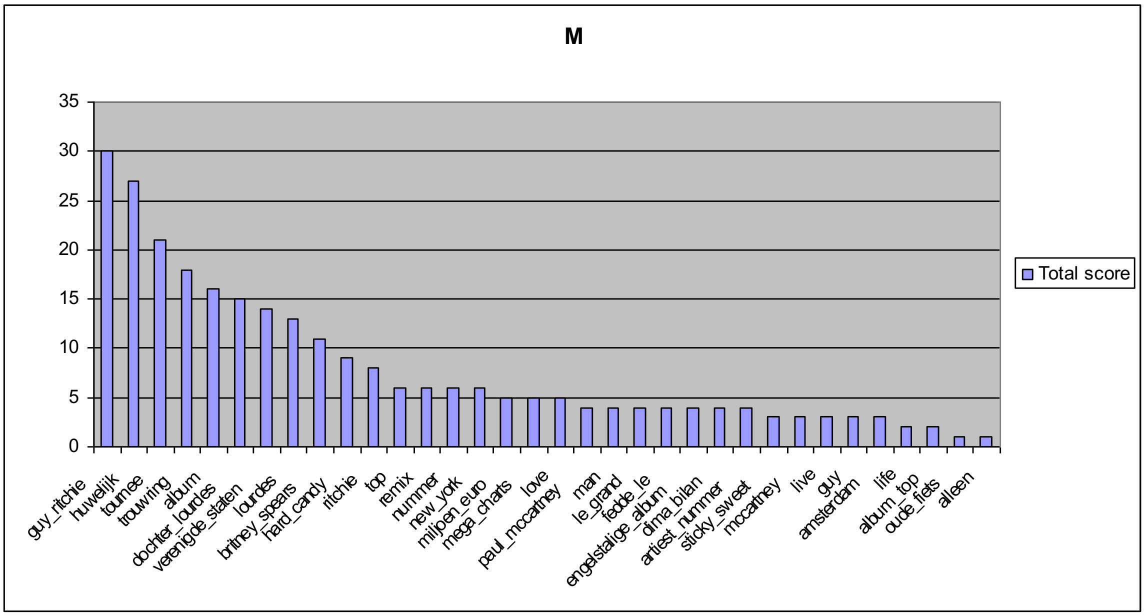

Figure A6: Scores for M

It is not obvious from Figure A6 that the word "dochter_lourdes" (score of 15) is preferred above the word "Lourdes" (score of 13). However, it is clear that the word "guy_ritchie" (score of 30) is more favored than the words "guy" (score of 3) and "ritchie" (score of 8) itself. Another word that is more preferred is "album" (score of 16) above "album_top" (score of 2) and top (score of 6). The word "mccartney" (score of 3) is not favored more or less than the word "paul_mccartney" (score of 4).

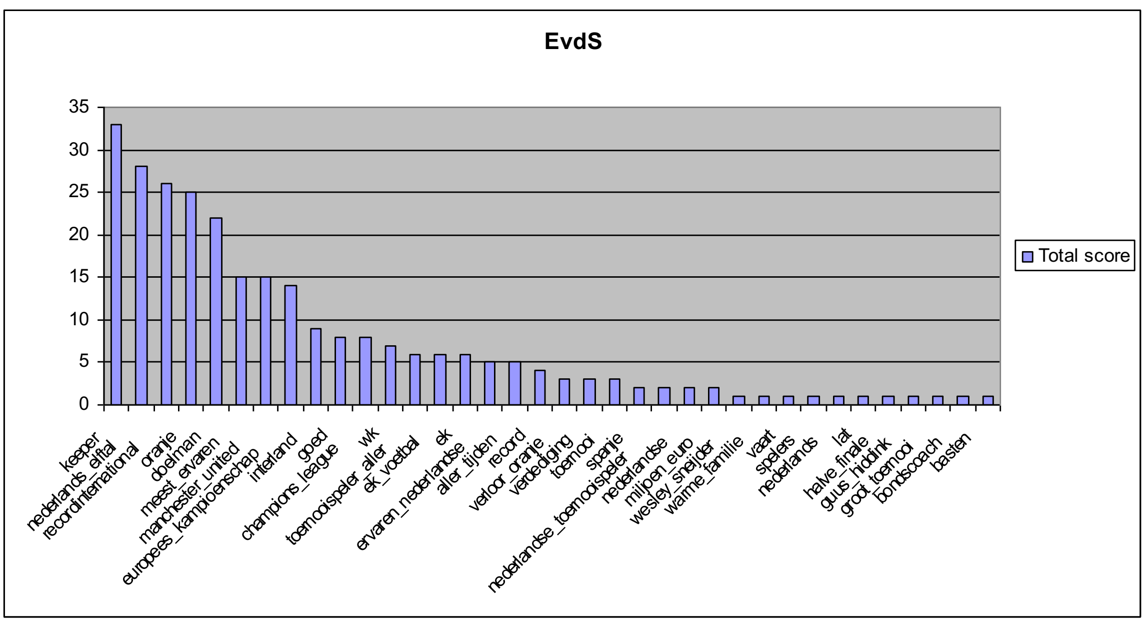

Figure A7: Scores for EvdS

It is not obvious from Figure A7 that the word "nederlands_elftal" (score of 28) is preferred above the word "oranje" (score of 26). However, it is clear that the word "keeper" (score of 33) is more favored than the word "doelman" (score of 22) itself. Another word that is more preferred is "europees_kampioenschap" (score of 14) above the word "ek" (score of 6). The word "groot_toernooi" (score of 1) is not favored more or less than the word "toernooi" (score of 3).

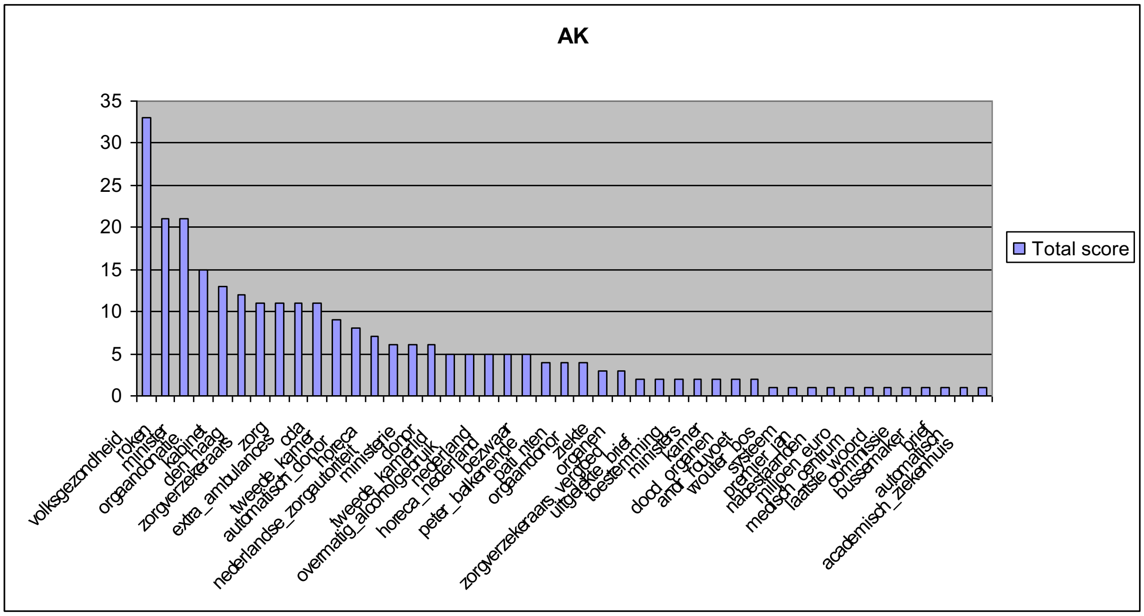

Figure A8: Scores for AK

It is obvious from Figure A8 that the word "orgaan_donatie" (score of 15) is preferred above the words "donor" (score of 6), "orgaan_donor" (score of 4), "dood_organen" (score of 2). However, it is clear that the word "automatisch_donor" (score of 8) is not more favored than the word "donor". The word "horeca_nederland" (score of 5) is not favored more or less than the word "horeca" (score of 7). A word like "zorgverzekeraars_vergoed" (score of 2) is less preferred than the word "zorgverzekeraars" (score of 11) itself.

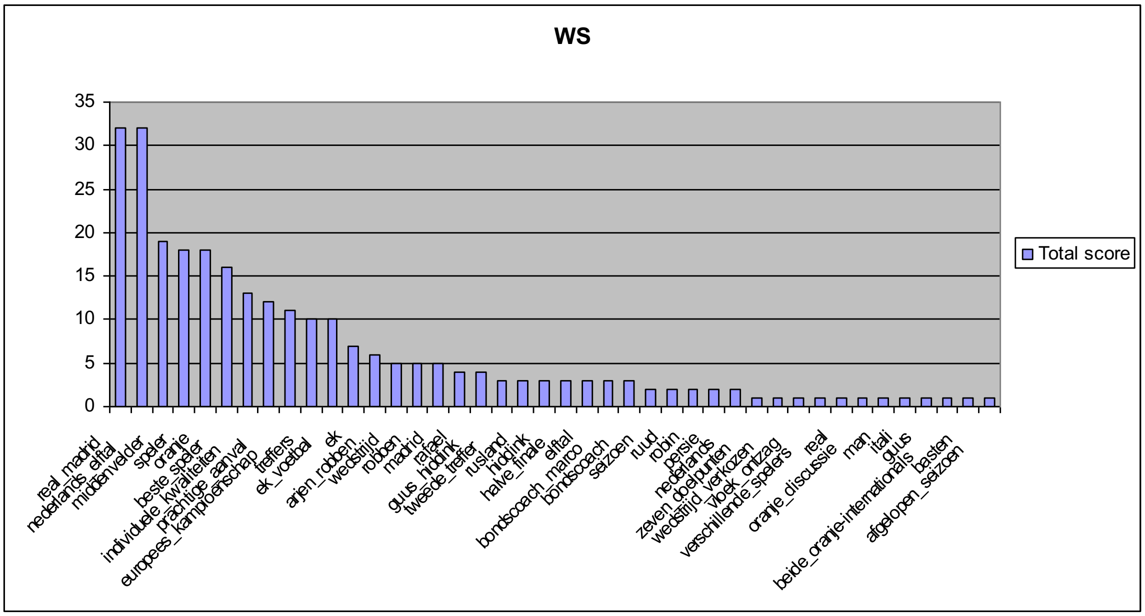

Figure A9: Scores for WS

It is obvious from Figure A9 that the word "nederlands_elftal" (score of 32) is preferred above the word "oranje" (score of 18). It is also clear that the word "real_madrid" (score of 32) is more favored than the word "madrid" (score of 5). The word "europees_kampioenschap" (score of 11) is not favored more than the word "ek" (score of 7). Another word that is not preferred more is "arjen_robben" (score of 7) over "robben" (score of 5). Both words "bondscoach_marco" and "bondscoach" have the same score of 3, so they are not favored above eachother. Also, for the words "speler" (score of 18) and "beste_speler" (score of 16) there is no obvious preference.

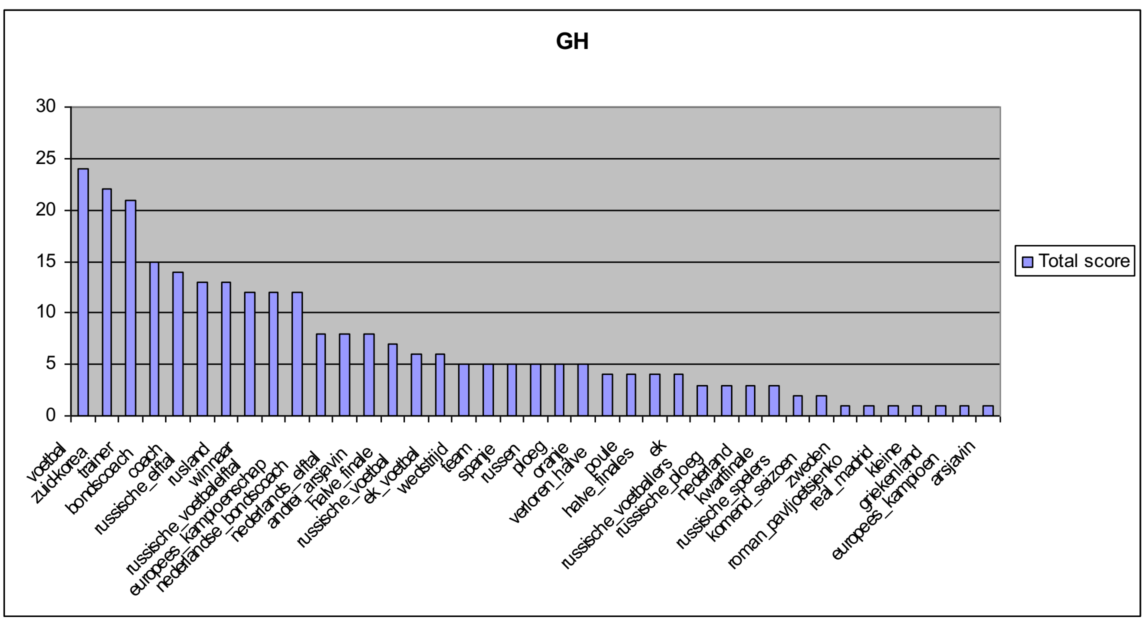

Figure A10: Scores for GH

It is obvious from Figure A10 that the word “bondscoach” (score of 15) is not more preferred than the word “coach” (score of 14). However, the word “bondscoach” is preferred above the word “nederlandse_bondscoach” (score of 8). It is also clear that the word “russisch_elftal” (score of 13) is not more favored than the word “russisch_voetbalelftal” (score of 12). The word “europees_kampioenschap” (score of 12) is preferred more than the word “ek” (score of 4). Another word that is not preferred more is “russische_voetballers” (score of 3) compared to “russische_ploeg” (score of 3) and “russische_spelers” (score of 2). The word “nederlands_elftal” (score of 8) has a slight preference over the word “oranje” (score of 5).

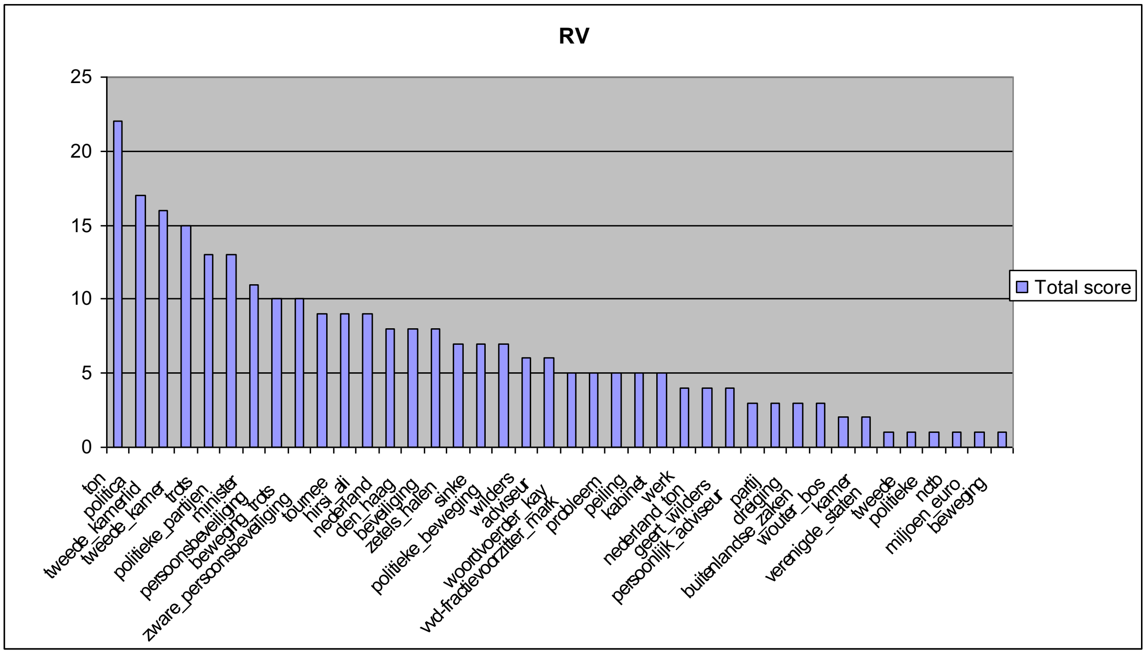

Figure A11: Scores for RV

It is obvious from Figure A11 that the word “persoonsbeveiliging” (score of 10) is not more preferred than the words “zware_persoonsbeveiliging” (score of 9) and “beveiliging” (score of 8). It is also clear that the word “trots” (score of 13) is not more favored than the word “beweging_trots” (score of 10). The word “ton” (score of 12) is preferred more than the word “nederland_ton” (score of 4). A word that is not preferred more is “wilders” (score of 6) compared to “geert_wilders” (score of 4). The word “tweede_kamer” (score of 15) has a high preference over the words “kamer” (score of 2) and “tweede” (score of 1). One more word that has a slight preference is “adviseur” (scoreof 6) compared to “persoonlijk_adviseur” (score of 3).

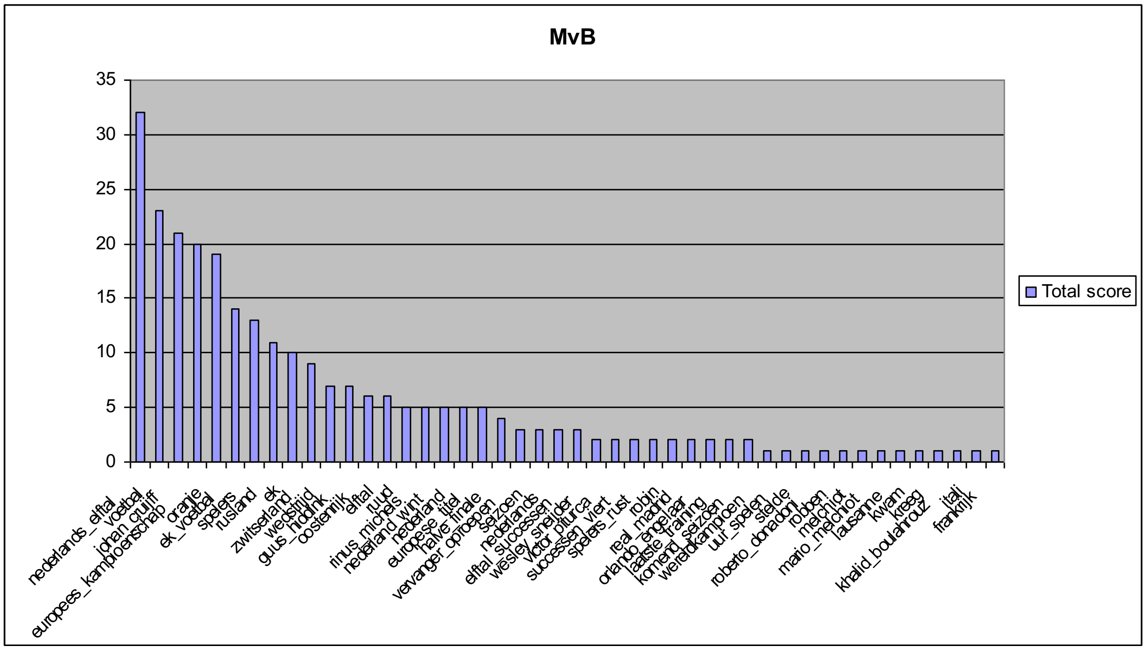

Figure A12: Scores for MvB

It is obvious from Figure A12 that the word “spelers” (score of 13) is more preferred than the word “spelers_rust” (score of 2). Also, the word “europees_kampioenschap” (score of 20) is preferred more than the word “ek” (score of 10). A word that is not favored more is “nederland_wint” (score of 5) above “nederland” (score of 5). The word “nederlands_elftal” (score of 32) has a high preference over the word “oranje” (score of 19).

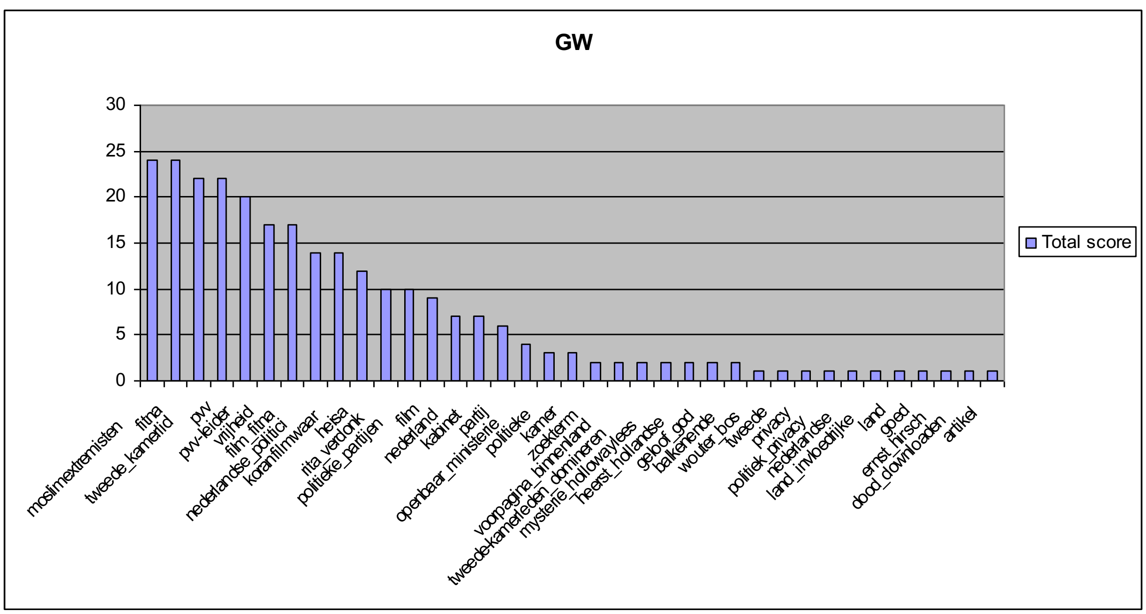

Figure A13: Scores for GW

It is obvious from Figure A13 that the word “fitna” (score of 24) is more preferred than the words “film_fitna” (score of 17) and “film” (score of 9).

H Detailed correlation coefficients

In this appendix the correlation coefficients are given when merging the 10 single words and the 10 two consecutive words to 20 composed words. This correlation is provided in Table A8.

Entity	Technique	Kendall's correlation	P-values
		CW (SW+TCW)	CW (SW+TCW)
PH	BoosTexter	0.16	0.129
	IG	0.07	0.502
	Oddsratio	-0.11	0.321
	Relief	0.38	0.000
	Ripper	0.15	0.147
	SVM	0.04	0.686
SD	BoosTexter	0.10	0.309
	IG	0.17	0.082
	Oddsratio	0.09	0.388
	Relief	0.08	0.440
	Ripper	0.17	0.080
	SVM	-0.08	0.424
BS	BoosTexter	0.08	0.487
	IG	0.20	0.074
	Oddsratio	0.06	0.582
	Relief	0.20	0.074
	Ripper	0.10	0.385
	SVM	-0.08	0.487
AA	BoosTexter	0.12	0.245
	IG	0.23	0.027
	Oddsratio	0.04	0.713
	Relief	0.17	0.104
	Ripper	0.32	0.002
	SVM	0.22	0.035
M	BoosTexter	-0.09	0.388
	IG	0.19	0.059
	Oddsratio	0.15	0.131
	Relief	0.09	0.392
	Ripper	-0.01	0.895
	SVM	0.00	0.989
EvdS	BoosTexter	0.00	0.969
	IG	0.22	0.028
	Oddsratio	0.04	0.671
	Relief	0.15	0.142
	Ripper	0.23	0.021
	SVM	-0.01	0.912

AK	BoosTexter	0.06	0.520
	IG	0.15	0.130
	Oddsratio	0.09	0.369
	Relief	0.25	0.010
	Ripper	0.08	0.412
	SVM	0.33	0.001
WS	BoosTexter	-0.12	0.229
	IG	0.07	0.463
	Oddsratio	0.16	0.101
	Relief	0.28	0.005
	Ripper	0.19	0.056
	SVM	-0.09	0.379
GH	BoosTexter	0.20	0.059
	IG	0.09	0.409
	Oddsratio	0.28	0.006
	Relief	0.02	0.844
	Ripper	0.18	0.082
	SVM	0.01	0.948
RV	BoosTexter	-0.10	0.335
	IG	0.17	0.089
	Oddsratio	0.08	0.402
	Relief	0.19	0.063
	Ripper	0.22	0.029
	SVM	0.06	0.532
MvB	BoosTexter	0.26	0.013
	IG	0.01	0.942
	Oddsratio	0.33	0.002
	Relief	-0.12	0.257
	Ripper	0.11	0.310
	SVM	-0.15	0.162
GW	BoosTexter	0.42	0.000
	IG	-0.08	0.415
	Oddsratio	0.05	0.599
	Relief	0.07	0.457
	Ripper	0.16	0.100
	SVM	0.05	0.578

Table A8: Kendall's correlation coefficient for 12 entities for two composed words

The correlations between humans and techniques by adding world knowledge is given in Table A9.

Entity	Technique	Kendall's correlation			P-values		
		SW	TCW	CW	SW	TCW	CW
PH	BoosTexter	0.12	0.18	0.19	0.414	0.269	0.094
	IG	0.26	0.06	0.10	0.084	0.719	0.370

	Oddsratio	-0.03	-0.08	-0.09	0.852	0.614	0.415
	Relief	0.41	0.26	0.39	0.006	0.113	0.000
	Ripper	0.058	0.26	0.17	0.741	0.108	0.118
	SVM	-0.01	0.08	0.06	0.952	0.614	0.599
SD	BoosTexter	0.21	0.01	0.12	0.138	0.985	0.226
	IG	-0.05	0.36	0.15	0.730	0.013	0.123
	Oddsratio	0.08	0.00	0.06	0.593	1.000	0.529
	Relief	0.03	0.13	0.09	0.849	0.382	0.378
	Ripper	0.25	0.12	0.20	0.070	0.403	0.049
	SVM	-0.02	-0.16	-0.07	0.890	0.262	0.515
BS	BoosTexter	0.03	0.38	0.13	0.878	0.023	0.300
	IG	0.28	0.46	0.29	0.111	0.006	0.016
	Oddsratio	0.03	0.33	0.14	0.906	0.049	0.254
	Relief	-0.11	0.47	0.20	0.528	0.004	0.093
	Ripper	0.19	0.27	0.16	0.274	0.108	0.183
	SVM	0.06	0.01	-0.04	0.759	0.955	0.750
AA	BoosTexter	0.10	0.14	0.12	0.496	0.389	0.245
	IG	0.24	0.25	0.23	0.094	0.107	0.027
	Oddsratio	-0.09	0.21	0.04	0.531	0.188	0.713
	Relief	0.12	0.23	0.17	0.407	0.147	0.104
	Ripper	0.37	0.19	0.32	0.009	0.249	0.002
	SVM	0.21	0.21	0.22	0.136	0.188	0.035
M	BoosTexter	-0.20	0.02	-0.06	0.165	0.893	0.576
	IG	0.04	0.33	0.17	0.810	0.023	0.115
	Oddsratio	0.14	0.22	0.18	0.345	0.144	0.089
	Relief	0.14	0.09	0.11	0.345	0.563	0.302
	Ripper	0.13	-0.09	0.01	0.365	0.538	0.963
	SVM	0.08	0.02	0.01	0.603	0.923	0.919
Evd S	BoosTexter	-0.01	-0.03	0.03	0.973	0.867	0.782
	IG	0.16	0.25	0.22	0.257	0.081	0.032
	Oddsratio	0.18	-0.08	0.03	0.204	0.603	0.801
	Relief	0.16	0.11	0.11	0.243	0.447	0.268
	Ripper	0.25	0.11	0.18	0.078	0.447	0.086
	SVM	0.16	-0.20	0.02	0.250	0.175	0.841
AK	BoosTexter	-0.05	0.20	0.07	0.729	0.175	0.489
	IG	0.35	-0.11	0.13	0.013	0.456	0.213
	Oddsratio	0.03	0.15	0.10	0.849	0.320	0.316
	Relief	0.28	0.22	0.27	0.046	0.141	0.007
	Ripper	0.02	0.18	0.10	0.903	0.221	0.339
	SVM	0.39	0.22	0.32	0.005	0.131	0.001
WS	BoosTexter	-0.15	-0.12	-0.10	0.286	0.392	0.326
	IG	-0.06	0.16	0.10	0.685	0.274	0.319
	Oddsratio	0.18	0.09	0.14	0.208	0.535	0.165
	Relief	0.18	0.31	0.27	0.196	0.030	0.009
	Ripper	0.24	0.05	0.22	0.089	0.715	0.033

	SVM	-0.23	0.04	-0.07	0.105	0.771	0.504
GH	BoosTexter	0.15	0.27	0.27	0.296	0.096	0.019
	IG	-0.04	0.19	0.18	0.794	0.245	0.112
	Oddsratio	0.32	0.27	0.35	0.028	0.092	0.002
	Relief	0.13	-0.05	0.07	0.366	0.751	0.527
	Ripper	0.00	0.29	0.22	1.000	0.077	0.055
	SVM	-0.14	0.03	0.03	0.345	0.875	0.769
RV	BoosTexter	-0.11	-0.06	-0.07	0.458	0.677	0.469
	IG	0.37	-0.03	0.17	0.011	0.871	0.092
	Oddsratio	0.04	0.15	0.11	0.815	0.294	0.291
	Relief	0.11	0.32	0.20	0.458	0.025	0.047
	Ripper	0.44	0.00	0.21	0.002	1.000	0.033
	SVM	0.19	-0.07	0.06	0.204	0.625	0.556
MvB	BoosTexter	0.28	0.17	0.24	0.052	0.271	0.028
	IG	0.10	-0.09	-0.01	0.518	0.566	0.905
	Oddsratio	0.34	0.25	0.30	0.019	0.097	0.005
	Relief	-0.17	-0.05	-0.13	0.267	0.754	0.224
	Ripper	0.10	0.09	0.08	0.524	0.541	0.476
	SVM	-0.17	-0.13	-0.12	0.255	0.396	0.244
GW	BoosTexter	0.37	0.52	0.41	0.007	0.000	0.000
	IG	0.03	-0.15	-0.05	0.825	0.302	0.625
	Oddsratio	0.28	-0.13	0.09	0.036	0.369	0.357
	Relief	0.14	0.06	0.11	0.298	0.688	0.262
	Ripper	0.15	0.16	0.10	0.286	0.293	0.313
	SVM	-0.05	0.22	0.02	0.717	0.146	0.808

Table A9: Kendall's correlation coefficient for 12 entities

11 Abbreviations

AA	Ahmed Aboutaleb
AK	Ab Klink
ANOVA	Analysis of variance
BOW	Bag of Words
BS	Britney Spears
Celebs	Celebrities
CW	Composed words
DL	Description length
EvdS	Edwin van der Sar
GH	Guus Hiddink
GW	Geert Wilders
IG	Information Gain
M	Madonna
MvB	Marco van Basten
PH	Paris Hilton
RV	Rita Verdonk
SD	Snoop Dogg
SVM	Support vector machine
SVM-Class	SVM as classifier
SVM-FS	SVM as Feature selection Technique
SW	Single words
TCW	Two consecutive words
WS	Wesley Sneijder

12 List of Tables

13 List of Figures

References

[1] Robert E. Schapire, Yoram Singer, *"BoosTexter: A Boosting-based System for Text Categorization"*, Machine Learning, 39(2/3): 135-168, 2000.

[2] Dunja Mladenic, Janez Brank, Marko Grobelnik, Natasa Milic-Frayling, "*Feature selection using Linear Classifier Weights: Interaction with Classification Models"*, The 12th National Conference on Artificial Intelligence, Pittsburgh, PA, USA, July 2005.
Link: research.microsoft.com/users/natasamf/publications/p181-mladenic.pdf

[3] http://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm

[4] Igor Kononenko and Marko Robnik-Sikonja, Uros Pompe, *"ReliefF for estimation and discretization of attributes in classification, regression, and ILP problems"*
Link: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.3929

[5] Igor Kononenko and Marko Robnik-Sikonja, *"Theoretical and Empirical Analysis of ReliefF and RReliefF"*, Machine Learning, Volume 53, Numbers 1-2, October 2003, Pages: 23-69(47)
Link: lkm.fri.uni-lj.si/rmarko/papers/robnik03-mlj.pdf

[6] Evgeniy Gabrilovich and Shaul Markovitch, *"Text Categorization with Many Redundant Features: Using Aggressive Feature Selection to Make SVMs Competitive with C4.5"*, In Proceedings of The Twenty-First International Conference on Machine Learning, pages 321-328, Banff, Alberta, Canada, 2004. Morgan Kaufmann
Link: www.cs.technion.ac.il/~gabr/papers/fs-svm.pdf

[7] http://www.ercim.org/publication/Ercim_News/enw62/novovicova.html

[8] Isabelle Guyon, Jason Weston, Stephen Barnhill and Vladimir Vapnik, *"Gene Selection for Cancer Classification using Support Vector Machines"*, Machine Learning, Volume 46, Issue 1-3, Pages: 389 - 422 , 2002
Link: www.cs.ust.hk/~qyang/bio/GuyonWBV-ML-2002.pdf

[9] Yiming Yang and Jan O. Pedersen, *"A Comparative Study on Feature Selection in Text Categorization"*, Proceedings of the Fourteenth International Conference on Machine Learning, Pages: 412 – 420, 1997

[10] David D. Lewis, *"Feature Selection and Feature Extraction for Text Categorization"*, Speech and Natural Language: Proceedings of the workshop held at Harriman, New York Pages: 212 – 217, Morgan Kaufmann Publishers, San Mateo, February 1992

[11] Zhaohui Zheng, Xiaoyun Wu, Rohini Srihari, *"Feature Selection for Text Categorization on Imbalanced Data"*, ACM SIGKDD, Volume 6, Issue 1, Pages 80-89, June 2004

[12] George Forman, *"An Extensive Empirical Study of Feature Selection Metrics for Text Classification"*, Journal of Machine Learning Research, Volume 3, Pages: 1289 – 1305, March 2003
Link: jmlr.csail.mit.edu/papers/volume3/forman03a/forman03a_full.pdf

[13] L. Douglas Baker and Andrew Kachites McCallum, *"Distributional Clustering of Words for Text Classification"*, Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, Melbourne, Australia, Pages: 96 – 103, 1998
Link: www.cs.cmu.edu/afs/cs.cmu.edu/user/mccallum/www/papers/**clustering-**sigir98s.ps.gz

[14] J.Novovičová, A.Malík , *"Text document classification using finite mixtures"*, *Research Report UTIA CAS*, No. 2063, December 2002
Link: library.utia.cas.cz/prace/20030016.ps

[15] Ron Bekkerman, Ran El-Yaniv, Naftali Tishby, Yoad Winter, *"Distributional Word Clusters vs. Words for Text Categorization"*, Journal of Machine Learning Research, Volume 3, Pages: 1183 - 1208, March 2003
Link: www.cs.technion.ac.il/~ronb/papers/jmlr.pdf

[16] George Forman, *"Feature Selection for Text Classification"*, Published as a book chapter in Computational Methods of Feature Selection, 2007

[17] Jihoon Yang and Vasant Honavar, *"Feature Subset Selection Using A Genetic Algorithm"*, Intelligent Systems and Their Applications, IEEE**,** Volume: 13, Issue: 2 Page: 44-49, 1998

[18] M. Dash, H. Liu, *"Feature Selection for Classification"*, Intelligent Data Analysis, Volume: 1, No. 3, Pages: 131-156, 1997
Link: www.public.asu.edu/~huanliu/papers/ida97.ps

[19] Igor Kononenko, Edvard Simec, "Induction of decision trees using ReliefF", In G. Della Riccia, R. Kruse, & R. Viertl (Eds.), Mathematical and Statistical Methods in Artificial Intelligence, CISM Courses and Lectures No. 363. Springer Verlag, 1995
Link: http://ai.fri.uni-lj.si/papers/kononenko94-issek.ps.gz

[20] Mark A. Hall, Lloyd A. Smith, *"Feature Subset Selection: A Correlation Based Filter Approach"*, In: Kasabov N., editor. Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Information Systems: New Zealand. Pages: 855–858 Springer; 1998.
Link: www.cs.waikato.ac.nz/~ml/publications/1997/Hall-LSmith97.pdf

[21] Wenqian Shang , Houkuan Huang , Haibin Zhu , Yongmin Lin , Youli Qu , Zhihai Wang, *"A novel feature selection algorithm for text categorization"*, Expert Systems with Applications: An International Journal, Volume 33 , Issue 1, Pages 1-5, 2007
Link: www.nipissingu.ca/faculty/haibinz/research/ESA07.pdf

[22] Jun Yan, Ning Liu, Benyu Zhang, Shuicheng Yan, Zheng Chen, Qiansheng Cheng, Weiguo Fan, Wei-Ying Ma, *"OCFS: Optimal Orthogonal Centroid Feature Selection for Text Categorization"*, Annual ACM Conference on Research and Development in Information Retrieval: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, Pages: 122 – 129, 2005
Link: research.microsoft.com/users/byzhang/publications/ocfs_f.pdf

[23] Michal Haindl, Petr Somol, Dimitrios Ververidis, Constantine Kotropoulos, *"Feature Selection Based on Mutual Correlation"*, CIARP06, Pages:569-577 Springer.

[24] Glenn Fung; O. L. Mangasarian, *"A Feature Selection Newton Method for Support Vector Machine Classification"*, Computational Optimization and Applications, Volume 28, Issue 2, Pages: 185 – 202, July 2004

[25] Fred S. Richardson, William M. Campbell, "Discriminative Keyword Selection using Support Vector Machines", NIPS (Neural Information Processing Systems Foundation), 2007.
Link books.nips.cc/papers/files/nips20/NIPS2007_0703.pdf

[26] Janez Brank, Marko Grobelnik, Nataša Milic-Frayling, Dunja Mladenic, *"Feature Selection Using Linear Support Vector Machines"*, Microsoft Technical Report MSR-TR-2002-63, June 2002

[27] Thorsten, Joachims, *"A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization"*, Proceedings of ICML-97, 14th International Conference on Machine Learning, 1997

[28] Thorsten, Joachims, *"Text categorization with support vector machines: Learning with many relevant features"*, LS8-Report 23, Universität Dortmund, LS VIII-Report, 1997
Link: www.cs.cornell.edu/people/tj/publications/joachims_98a.ps.gz

[29] James Tin-yau Kwok, *"Automated text categorization using support vector machine"*, In Proceedings of the International Conference on Neural Information Processing, Pages: 347-351, 1999

[30] Erik Wiener, Jan O. Pedersen and Andreas S. Weigend, *"A neural network approach to topic spotting"*, In: Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR), 1995

[31] David D. Lewis and Marc Ringuette, "*A comparison of two learning algorithms for text categorization*", In Third Annual Symposium on Document Analysis and Information Retrieval, Pages 81-93, 1994

[32] Yiming Yang, *"An evaluation of statistical approaches to text categorization"*, Information Retrieval 1, Pages 69–90, 1999

[33] Lei Yu and Huan Liu, *"Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution"*, In Proceedings of The Twentieth International Conference on Machine Leaning (ICML-03), Pages 856-863, Washington, D.C., August 21-24, 2003
Link: www.hpl.hp.com/conferences/icml2003/papers/144.pdf

[34] Ron Kohavi and George H. John, *"Wrappers for feature subset selection"*, Artificial Intelligence, Volume 97, Issue 1-2, Special issue on relevance, Pages: 273 – 324, December 1997
Link: robotics.stanford.edu/~ronnyk/wrappers.ps

[35] Sammy Das, "*Filters, Wrappers and a Boosting-Based Hybrid for Feature"*, Proceedings of the Eighteenth International Conference on Machine Learning, Pages: 74 – 81
Link: www.cs.rpi.edu/~sanmay/papers/icml01.ps.gz

[36] William W. Cohen, *"Fast effective rule induction"*, Machine Learning: Proceedings of the Twelfth International Conference (ML95)
Link: www.cs.cmu.edu/~wcohen/postscript/ml-95-ripper.ps

[37] Jihoon Yang, Asok Tiyyagura, Fajun Chen, Vasant Honavar, *"Feature Subset Selection for Rule Induction Using RIPPER",* Proceedings of the Genetic and Evolutionary Computation Conference, 2, Page1800. Orlando, Florida, USA, Morgan Kaufmann, 13-17 July1999
Link: www.cs.bham.ac.uk/~wbl/biblio/gecco1999/RW-738.ps

[38] www-users.cs.umn.edu/~kumar/dmbook/dmslides/chap5_alternative_classification.ppt

[39] www.csie.ntnu.edu.tw/~violet/ML95/chap9.ppt

[40] Ian H. Witten and Eibe Frank, *"Data Mining: Practical Machine Learning Tools and Techniques, second edition"*, 2005 by Elsevier.

[41] http://www.statsdirect.com/help/nonparametric_methods/kend.htm

[42] De Gunst, M.C.M. (najaar 2006), "Statistical Models", Lecture notes VU Amsterdam.

[43]
http://www.rtl.nl/(/actueel/editienl/)/components/actueel/editienl/2006/16/woordenschat.xml

[44] http://nlp.stanford.edu/IR-book/html/htmledition/soft-margin-classification-1.html