May 2014

Universiteit Leiden
Opleiding Informatica

Automated Multi-Label Classification
of the Dutch Speeches from the Throne

Wouter Eekhout

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

For social science researchers, qualitative content analysis and classification is a time consuming and costly endeavor. The Dutch Speeches from the Throne (1945–2010), for example, constitute a data set of 9143 sentences and 11779 unique words. Manually labeling these sentences is a cumbersome task.

As part of the Political Attention Radar this thesis presents an automated Dutch multi-label classification system. The presented solution uses a traditional bag-of-words document representation, an extensive set of human coded examples, and an exhaustive topic coding system to automatically classify each sentence into none, one or several categories. For social science researchers this thesis contributes a new view on the data through an automated classification system. For computer science it contributes a multi-labeling approach to existing classification solutions.

Contents

Chapter 1

Introduction

Since 1995 the techniques and capacities to store new electronic data and to make it available to many persons have become a common good. As of then, different organizations, such as research institutes, universities, libraries, and private companies (Google) started to scan older documents and make them electronically available as well. This has generated a lot of new research opportunities for all kinds of academic disciplines.

The use of software to analyze large data sets has become an important part of doing research in the social sciences. Most academics rely on human coded data sets, both in qualitative and quantitative research. However, with the increasing amount of data sets and the complexity of the questions scholars pose to the data sets, the quest for more efficient and effective methods is now on the agenda.

In this document research is described for automated multi-label classification of Dutch data sets. This research is a master thesis for the course Master Computer Science at LIACS, under the supervision of Walter Kosters and Christoph Stettina. LIACS is the computer science institute of Leiden University. The focus lies on the Dutch Speeches from the Throne data set. The thesis is written for people who have a basic knowledge about classifiers. Some examples and problems are specific for the Dutch language.

Chapter 2 is about the background for this research and the research question. In Chapter 3 the related research is described. Chapter 4 gives a in-depth view of the data set. Chapter 5 provides an overview of working order of the classifiers for Chapter 6 (preprocessing of the data set) and Chapter 7 (the classifiers). Finally in Chapter 8 the results are described. These results are discussed in Chapter 9 along with future work.

Chapter 2

Background and research question

Agenda-setting is essential to the policy process, it sets the stage for which problems are attended to and which are ignored, and how these problems are portrayed and prioritized. The underlying idea is that attention is a scarce resource, and this means that agendas of governments, parliaments, and other venues are always selective and are the target of pressure. Since 2009, the team of Campus the Hague at the Montesquieu Institute studies political agendas in the Netherlands, other democratic countries, and the European Union in order to understand how problems are addressed over a long period of time. This research is done in close collaboration with an international network of scholars in Europe and North America, under the label of the Comparative Agendas Project[1]. One of their activities is mapping policy agendas in the Netherlands and the EU. These agendas can be considered separately or be compared to see how much attention major themes of public policy receive in different institutional venues. One of the data sets comprises of the the annual political agenda of Dutch governments since 1945, the so-called Speeches from the Throne delivered at the beginning of the budgetary year, the third Tuesday of September. The data set is constructed by manually content coding all statements, using a common policy topic classification system, with 19 major topics (such as macroeconomics, foreign affairs, health, etc.) and more specific subtopics for each of the major categories. The coding system is unified and used by all member teams in the Comparative Agendas Project. For social science researchers, content analysis and classification of the data set has been time consuming and costly.

Problem The rapid growth of digitized government documents in recent

[1] `www.comparativeagendas.org`

years presents new opportunities for research but also new challenges. With more and more data coming online, relying on human annotators becomes prohibitively expensive for many tasks. Classifying the data set takes a lot of time and effort.

Goal/Project summary An automated multi-label classification system based on the Dutch data set that consists of the Speeches from the Throne from 1945–2010.

Scope The automated classification system will only be designed for texts that are in the Dutch language and contain political subjects. The system will only classify the text. To classify texts, a classified data set will be used.

Chapter 3

Related research

Due to the multi-disciplinary nature of this project, crossing the domains of social studies and computer science. This chapters addresses the current work done of topic classification in both domains. When it comes to related research there are two types; social studies related researches and computer science related researches. The computer science related research focuses on automated classifiers, where the social studies related research focuses on getting relevant information out of data.

3.1 Social studies

Qualitative research is a method of inquiry employed in many different academic disciplines, traditionally in the social sciences, but also in market research and further contexts [37]. Qualitative researchers aim to gather an in-depth understanding of human behavior and the reasons that govern such behavior. The qualitative method investigates the why and how of decision making, not just what, where, when. Hence, smaller but focused samples are more often used than large samples. In the conventional view, qualitative methods produce information only on the particular cases studied, and any more general conclusions are only propositions (informed assertions). Quantitative methods can then be used to seek empirical support for such research hypotheses. The most common method is the qualitative research interview, but forms of the data collected can also include group discussions, observation and reflection field notes, various texts, pictures, and other materials.

Computer Assisted/Aided Qualitative Data Analysis Software (CAQDAS) offers tools that assist with qualitative research [36]. CAQDAS is used in psychology, marketing research, ethnography, and other social sciences. The CAQDAS Networking project[1] lists the following tools. A CAQDAS program should have:

- Content searching tools
- Coding tools
- Linking tools
- Mapping or networking tools
- Query tools
- Writing and annotation tools

There is one paper that use the same data set [2]. The provided systematic examination of the expectation that coalition continuity results in relative agenda stability by analyzing agenda-setting patterns in the Netherlands over the post-WWII period as represented in the annual Speeches from the Throne to the Parliament. The paper examines the nature of Queens speeches and the Dutch agenda process and discuss the coding procedures used to dissect the speeches in terms of policy content. The main part of the paper discusses the macro-structure of the policy agenda reflected in the speeches over the post-WWII period with emphasis on the changes in the patterns of attention associated with changes in governments. The conclusion is that newly appointed governments only modestly change the distribution of attention for major policy topics, and entirely new coalitions, in most cases, even seem relatively averse of redirecting political attention in their first Speech from the Throne, while in the Dutch institutional context an "entirely new coalition" never involves a complete coalition turnover.

The paper [20] has done exactly the same study, but for the American bills. The categories are the same as the Dutch ones, even the numbering of the categories. The chosen two-phase hierarchical approach to Support Vector Machine (SVM) training which mimics the method employed by human coders. It begins with a first pass which trains a set of SVM's to assign one of 20 major topics to each bill. The second pass iterates once for each major topic code and trains SVM's to assign subtopics within a major class. For example, take all bills that were first assigned the major topic of health (3) and then train a collection of SVM's on the health subtopics (300–398). Since there are 20 subtopics of the health major topic, this results in an additional 20 sets of SVM's being trained for the health subtopics. Once the SVM's have been trained, the final step is subtopic selection. In this step, assessing the predictions from the hierarchical evaluation to make the best guess prediction for a bill. For each bill, apply the subtopic SVM classifiers from each of the top

3 predicted major topic areas (in order to obtain a list of many alternatives). This gives a subtopic classification for each of the top 3 most likely major categories. The system can then output an ordered list of the most likely categories for the research team.

3.2 Automated classification

The most related research is the paper [3]. This paper uses the same data set, with only one minor difference: this paper has the speeches from the throne from 1945–2008. The paper combines the results of the SVM, Maxent and LingPipe classifiers and explores additional ways of mingling the various categorization methods and algorithms. These methods and algorithms are supervised solutions. They rely on human analysts. Human analysts should code a large number of texts, to inform algorithms for supervised machine learning about typical features of texts that belong to the various categories of the category system, so to enable classification of new texts.

The paper [21] investigates classification of emails sent by the political parties during the 2004 presidential election. Given an email without contextual information, it classifies it as originating from either the Republican or Democratic Party. While this type of task is well practiced in the political communication literature, it uses this exercise to demonstrate a new methodological technique to bridge the qualitative world of coding and context analysis with empirical analysis and methods. The experiment involves two parallel studies using the same data set and coding rules. The first study is a traditional context analysis experiment. The second is a computer-assisted context analysis conducted. The focus of this paper is to describe how a skilled computer scientist would approach the problem of categorizing thousands of email messages. Text categorization problems are frequently encountered by political communication analysts, and current methods employ manual techniques or computer software which searches for keywords in the text.

The paper [27] proposes the following methodology and architecture; clustering for grouping and storing data. For text classification it suggests decision tree, k nearest neighbor, Naive Bayes and Support Vector Machines.

Clustering Clustering is defined as a process of grouping data or information into groups of similar types using some physical or quantitative measures [18]. These quantitative measures are based on different distance functions measured from a point called the centroid of the cluster. Different clustering techniques were tested to find the similarities between terms and the k means clustering technique was found to be good for dividing the useful information into multiple subspaces. k means clustering was ultimately used to discover the natural relationships between terms to further capture an initial level of knowledge. The similarities between terms are measured on the basis of Euclidean distance.

Decision Tree Analysis Decision tree analysis algorithms are most useful for classification problems and the process of building a decision tree starts with the selection of a decision node and splitting it into its sub nodes or leafs. A decision tree algorithm is Quinlan's algorithm C4.5 which generates decision trees [22] based on splitting each decision node by selection of a split and continuing its search until no further split is possible. It uses the concept of information gain [28] or entropy reduction to select the best split.

k **Nearest Neighbor Algorithm** The k nearest neighbor (K-NN) algorithm is a technique that can be used to classify data by using distance measures. It assumes the training set includes not only the data in the set but also the desired classification for each item. In effect, the training data becomes the model. The K-nearest neighboring algorithm works on the principle of finding the minimum distance from the new or incoming instance to the training samples [13]. On the basis of finding the minimum distance only the K closest entries in the training set are considered and the new item is placed into the class which contains the most items of the K closest items. The distance between the new or incoming item to the existing one is calculated by using some distance measure, and the most common distance function is the Euclidean distance.

Naive Bayes Algorithm A Naive Bayes algorithm is a simple and well-known classifier which is used in solving practical domain problems. The Naive Bayes classifiers are used to find the joint probabilities of words and classes within a given set of records [40]. This approach is based on Bayes' Theorem. It is assumed that classes are independent of each other and this is called the Naive assumption of class conditional independence and it is made while evaluating the classifier. The classification task is done by considering prior information and likelihood of the incoming information to form a posterior probability model of classification.

Support Vector Machines SVM's are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis [7]. Given a set of training examples, each marked as belonging to one of two categories, a SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap it falls on.

In all the mentioned papers regarding classifiers a word weighting algorithms has been used. Word weighting algorithms are used to measure the importance of a word. The paper [16] points out that there are three types of word weighting algorithms:

1. Structure based
2. Content based
3. Structure and content based

The distinction between the two is that the structure base weighting approach does not require any advanced knowledge of the potential significance. It uses a sophisticated approach to measure a importance of a weight. It uses a combination of the structural elements (context and/or linguistic structure of the sentence), the Part of Speech (is it a noun, verb, etc.), the complete word, the stem of a word and hypernyms of the word. Here the Structural elements were given a static low weight of 1. The Part of Speech nodes were given a static weight of 10, words were weighted according to their frequency in the data set using the TF-IDF (Term Frequency-Inverse Document Frequency) method. Stems were half the value of the Token and hypernyms one quarter the value. It suggests to use the following steps when cleaning up the words and selecting words: (i) word stem, (ii) word Part Of Speech (POS (e.g. DT, JJ, NN)), (iii) word order, (iv) word hypernyms [32], (v) sentence structure, (vi) sentence division and (vii) sentence order. The paper also uses the following techniques, as alternatives for the TF-IDF method:

- Pearson Correlation Coefficient [24] for structure based weighting.
- Node weight for content based node weighting (edge weighting would operate in a similar manner).
- Mutual information [21] for content based edge weighting.
- CHI (x^2) for content based class label discrimination weighting.
- Node entropy for combined structure and content based node weighting.

3.2.1 Multi-label classification

Most information on classification is meant for single classification. Usually taking the highest scoring result in the prediction. Not much research has been done in the field of multi-classification. In the paper [38], a Naive Bayes multi-label classification algorithm is proposed. It measures a minimum threshold score that is needed for the future predictions. Any result scoring higher than or equal to the threshold score gets returned as a prediction. In the paper [45], a multi-label lazy learning approach named multi-label k nearest neighbor is presented, which is derived from the traditional k Nearest Neighbor (kNN) algorithm. In detail, for each input, its k nearest neighbors in the training set are firstly identified. After that, based on statistical information gained from the label sets of these neighboring instances, i.e., the number of neighboring instances belonging to each possible class, the maximum a posteriori (MAP) [11] principle is utilized to determine the label set for the input. Another paper [26] maps documents together that have the same labels. For instance a document with label 3 and 15 gets into a bag labeled 3 and 15, so no overlap can occur in the bags. A similar approach has been tried in [12].

However in all the papers the discussed classifiers do not have very good results, meaning 90% accuracy or higher. It seems to be that there is more research needed regarding multi-label classification. Traditional single-label classification is concerned with learning from a set of examples that are associated with a single label from a set of disjoint labels [26]. The learning problem is called a binary classification problem (or filtering in the case of textual and web data). Taking the highest scoring result as the prediction is the standard method.

In multi-label classification, the examples are associated with a set of labels. In the past, multi-label classification was mainly motivated by the tasks of text categorization and medical diagnosis. Text documents usually belong to more than one conceptual class. For example, a newspaper article concerning the reactions of the Christian church to the release of the Da Vinci Code film can be classified into the categories Society, Religion and Arts, Movies. Similarly in medical diagnosis, a patient may be suffering for example from diabetes and back pain at the same time.

Nowadays, it seems that multi-label classification methods are increasingly required by modern applications, such as protein function classification [44], music categorization [19] and semantic scene classification [1]. In semantic scene classification, a photograph can belong to more than one conceptual class, such as sunsets and beaches, at the same time. Similarly, in music categorization a song may belong to more than one genre. For example, several hit songs of the popular rock band Scorpions can be characterized as both rock and ballad.

Chapter 4

The data set

As stated before the data set contains out of coded Dutch Speeches from the Throne [33] from the years 1945 to 2010. One speech from the throne is given per year. The data set consists of a total of 9143 sentences. These sentences contain 11779 unique words. The data set contains sentences that are classified into none, one or multiple political categories, such as “Macroeconomics”, “Foreign Trade”, “Government Operations”, “Health”, “Agriculture”, “Education”, “Transportation”, etc. There are 19 main categories and 20 main categories if you count “no category” as a separate category. The category identifier (id), category name and sentence count of these categories are documented in Table 4.1. When a sentence has been categorized with multiple categories the count for every categorized category gets one higher, respectively. What can be seen in the table is that the size of each category differs. The biggest category is category 19 “International Affairs and Foreign Aid” containing 1385 sentences. And the smallest category category 8 ”Energy” only contains 105 sentences. All the main categories have subcategories. For instance the main category ”Healthcare” has, among others, the subcategories “Medical-ethical issues”, “Healthcare reform” and “Waiting lists”.

These sentences has been categorized by hand by trained coders. Each speech from the throne has been categorized by at least two coders individually. After individually categorizing each sentences the results are compared. If the results differ too much the Speeches of the Throne are re-categorized. If not, the categories per sentence are accepted. The coders may differ per year. The categorization is based on a topic coding book [3], which was originally developed by Baumgartner and Jones and updated by Wilkerson and Adler in 2006[1]. It contains 19 main topics and 225 subtopics (8 extra topics have been added for media coding). Since 2000 various European scholars have

[1] http://www.policyagendas.org

Id	Name	Count
1	Macroeconomics	1190
2	Civil Rights, Minority Issues, and Civil Liberties	328
3	Health	284
4	Agriculture	236
5	Labor, Employment, and Immigration	700
6	Education	500
7	Environment	224
8	Energy	105
10	Transportation	308
12	Law, Crime, and Family Issues	420
13	Social Welfare	598
14	Community Development and Housing Issues	318
15	Banking, Finance, and Domestic Commerce	235
16	Defense	405
17	Space, Science, Technology, and Communications	158
18	Foreign Trade	184
19	International Affairs and Foreign Aid	1385
20	Government Operations	605
21	Public Lands and Water Management	169
0	(no label)	791

Table 4.1: The identifiers, names and sentence count of the labels

started to code their national data too, using the same code book, although they made minor adjustments to meet the country specifics. By now, there are teams coding in Belgium, Canada, Denmark, England, France, Italy, The Netherlands, Spain, Switzerland, and the United Kingdom. In addition to the national projects, some are also starting to code EU activities, such as the COM documents (EU directives) and EU parliamentary questions. The common code book that all countries use makes an easy comparison of policy attention between the different countries possible. The country teams meet regularly to exchange information and coordinate comparative projects. One of the informal agreements is to categorize the data sources at least back to 1978 (further, if possible). The teams also coordinate what and how to do the coding. The bills are, for instance, coded per bill, but the government agreements per section and the yearly opening speech of parliament per quasi-sentence.

A sample from the given data set is shown in Table 4.2. As can be seen, each sentence contains the category id and the year of the Speech from the

Category id	Sentence	Year
13	Alle kinderen moeten gelijke kansen krijgen om zich te kunnen ontwikkelen.	2010
6	Schooluitval moet effectief worden bestreden.	2010
2	Het actief beveiligen en beschermen van de samenleving en de burger tegen intimidatie, discriminatie en geweld blijft een hoge prioriteit van de regering.	2010
2	Het actief beveiligen en beschermen van de samenleving en de burger tegen intimidatie, discriminatie en geweld blijft een hoge prioriteit van de regering.	2010
12	Het actief beveiligen en beschermen van de samenleving en de burger tegen intimidatie, discriminatie en geweld blijft een hoge prioriteit van de regering.	2010

Table 4.2: Sample from the given data set

throne. All the sentences are in chronological order. Note that when a sentence has been categorized with multiple categories that sentence is split up into different rows.

The distribution of the categories differs from one sentence to another in a Speech from the Throne. When looking at the distribution of the categories no clear pattern exists in the distribution of the categories, as can been seen in Figure 4.1. On the vertical axis the category shown and on the horizontal axis the sentences in chronological order. The categories are scattered from the first sentence to the last. The category distribution from a different year looks completely different and just as scattered. For an overview of the category distribution compared to other years see Figure 4.2. Each sentence has a specific color. Each category stand for one category, as can been seen in the legenda. With this figure a better comparison of the category distribution of the several Speeches from the Throne is possible. The only similarity that most of the Speeches from the Throne have is that almost all start with category 0.

All the speeches from the throne (uncategorized) can be found on a website[2] with the paragraph tags `<p> ... </p>`. All the information within a paragraph can be important.

One word in a sentence can be enough for a categorization of that sentence.

[2] http://www.troonredes.nl

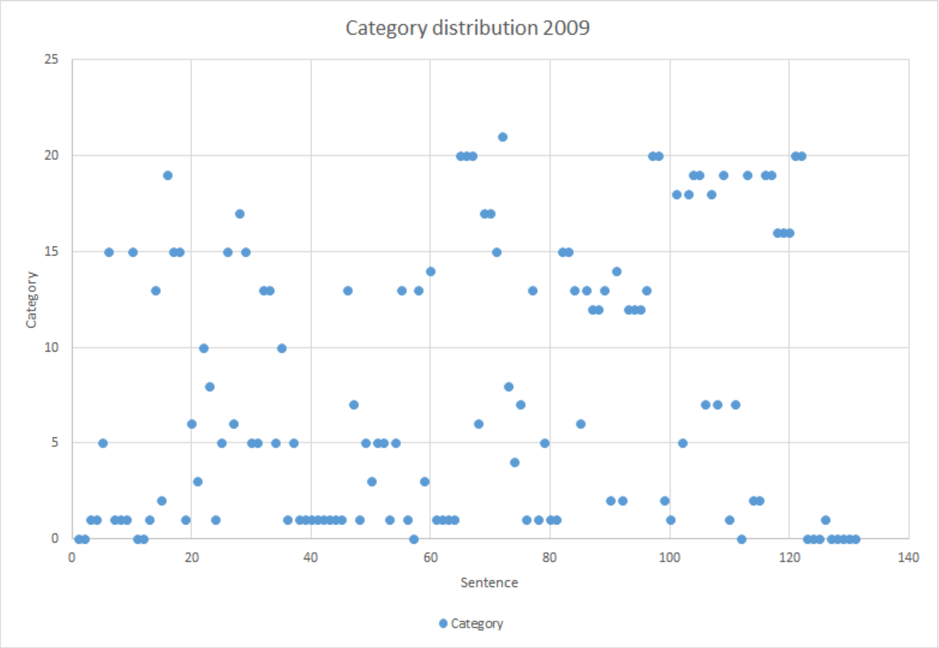
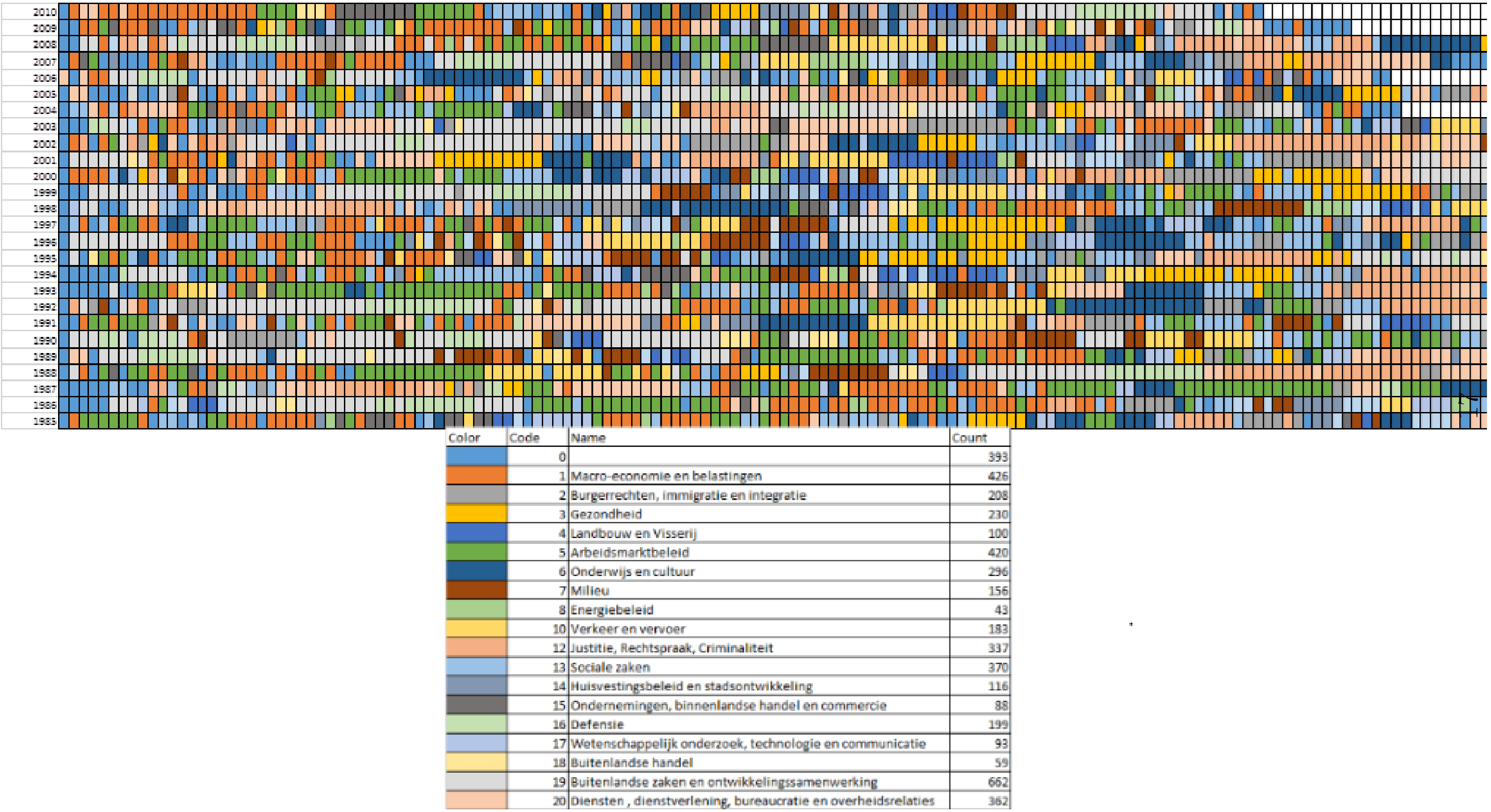


Figure 4.1: Category distribution of the year 2009

Sentences that often are within multiple categories have been categorized by individual words. One to three categories are often dominating per year, depending on the focus and the events in the past regarding the Kingdom of the Netherlands. Countries occur only within the categories “Foreign affairs” and/or “Defense”, even the (former) colonies of the Kingdom of the Netherlands such as “Nederlandse Antillen”. There is a different “tone” between the speeches. It is clear that per ruler and writer the speeches differ greatly. The word “pensioenuitkering” only occurs in the year 2010. Throughout the data set words regarding “pensioen” are categorized as 5 and 13, but not in all cases in both categories. Sentences are a reference to an earlier sentence, basically further explaining an earlier sentence. For instance take the two sentences: “Less new recruits will be recruited this year for the army. Even less than previous year.”. The first sentence is clearly about the army (category 16 Defense). The second sentence is a reference to the previous sentence, but the word army does not appear in it. So the classifier will probably not know if it is about the army, unless information from the surrounding sentences is used.



Color	Code	Name	Count
🟦	0		393
🟧	1	Macro-economie en belastingen	426
⬜	2	Burgerrechten, immigratie en integratie	208
🟨	3	Gezondheid	230
🟦	4	Landbouw en Visserij	100
🟩	5	Arbeidsmarktbeleid	420
🟦	6	Onderwijs en cultuur	296
🟫	7	Milieu	156
🟩	8	Energiebeleid	43
🟨	10	Verkeer en vervoer	183
🟧	12	Justitie, Rechtspraak, Criminaliteit	337
🟦	13	Sociale zaken	370
🟦	14	Huisvestingsbeleid en stadsontwikkeling	116
⬛	15	Ondernemingen, binnenlandse handel en commercie	88
🟩	16	Defensie	199
🟦	17	Wetenschappelijk onderzoek, technologie en communicatie	93
🟨	18	Buitenlandse handel	59
⬜	19	Buitenlandse zaken en ontwikkelingssamenwerking	662
🟧	20	Diensten , dienstverlening, bureaucratie en overheidsrelaties	362

Figure 4.2: Category distribution compared for different years

Chapter 5

Overview

This chapter contains the overview of the system, see Figure 5.1. It starts with a categorized data set, in our case the Speeches from the Throne. The data set gets preprocessed. During the preprocessing stage all the stop words are removed and stemming is applied as described in Section 6.1. What is left of every sentence is put into a bag of words. There are 20 bags of words, one for every category. The sentence goes into the corresponding bag based on the category. For each word in every bag the weight is calculated as described in Section 6.2, creating information about the importance of every word per category.

When a new sentence comes in as input it first gets to the preprocessing stage, the stop words are removed and stemming is applied. Also the text is split into sentences based on the dot, exclamation mark and the question mark. Depending on the classifier some other steps can be taken, such as calculating the word weights per word per sentence, see Chapter 7 for more information.

Now the system has the preprocessed data set and the preprocessed sentences from the input text. Each preprocessed sentence will be referenced against the preprocessed data set using one of the classifiers in Chapter 7. That classifier creates predictions for that sentence.

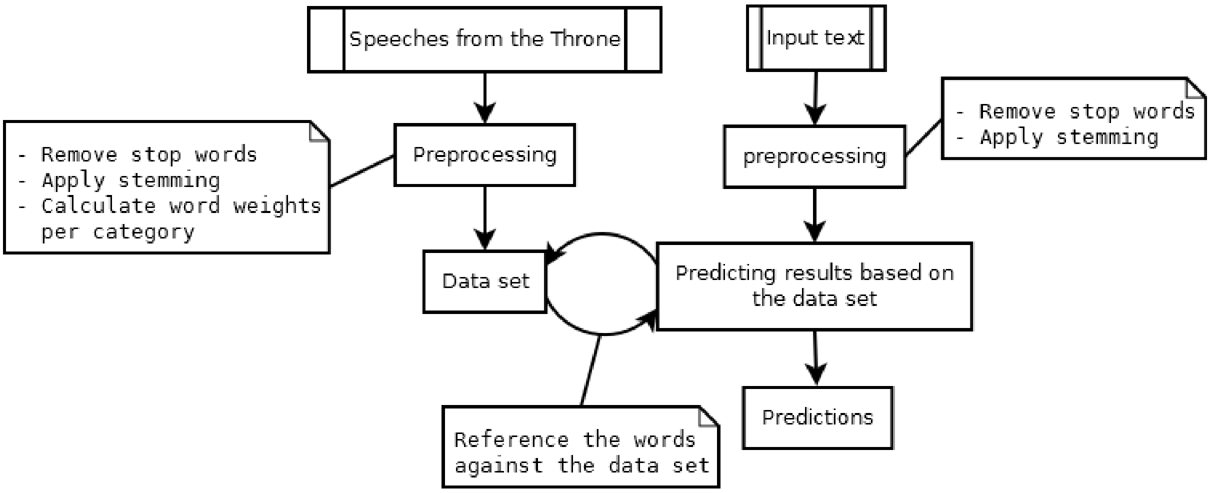


Figure 5.1: General working order of the classifier

Chapter 6

Preprocessing

The preprocessing is done is several stages. First the data set is prepared and then the word weights are calculated. Preparing is done by cleaning the text (removing non-words and apply stemming) and splitting the text into sentences. Then the weight of every word is calculated based on the information in all the documents.

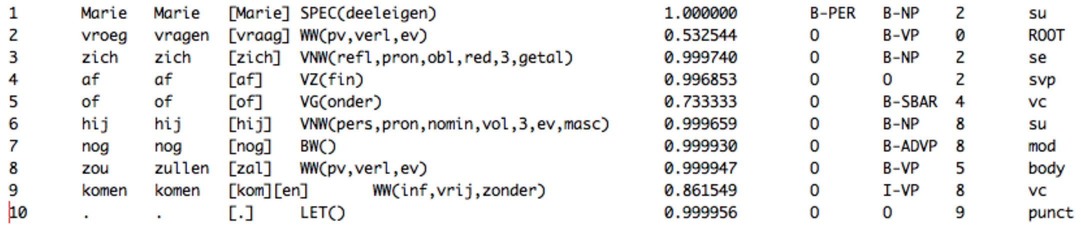
6.1 Preparing the data set

Firstly all the Dutch stop words are removed, non-tokens are removed and converted. The text is lowered case [20]. In Table 6.1 the Dutch stop words are documented. This reduces the amount of data and improves the speed. The stop words contain the word: “jaar”. This word is not an official stop word. However this word occurs in the top 10 most occurring words within the data set and it occurs in every category. Therefor this word is added to the stop words list, because for the Speeches of the Throne this word is a stop word. After removing the stop words the text is normalized.

For normalizing the text stemming or lemmatization can be used. Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. If confronted with the token English “saw”, stemming might return just s, whereas lemmatization would attempt to return either “see” or “saw” depending on whether the use of the token was as a verb or a noun. The two may also differ in that stemming most commonly collapses derivationally related words, whereas lemmatization commonly only collapses the different inflectional forms of a lemma. Linguistic processing for stemming or lemmatization is often done by an additional plug-in component to the indexing process, and a number of such components exist, both commercial and open-source. The most common algorithm for stemming English is Porter's algorithm [39]. However, this is designed for the English language [30].

The lemmatizer used for this research is Frog [28, 31]. Frog, formerly known as Tadpole, is an integration of memory-based natural language processing (NLP) modules developed for Dutch. All NLP modules are based on Timbl, the Tilburg memory-based learning software package. Frog's current version can tokenize, tag, lemmatize, and morphologically segment word tokens in Dutch text files, can assign a dependency graph to each sentence, can identify the base phrase chunks in the sentence, and can attempt to find and label all named entities, to a certain extent. The output can as shown in Figure 6.1, in order of the columns, has:

- Token number (resets every sentence)
- Token
- Lemma
- Morphological segmentation
- Part of Speech (PoS) tag
- Confidence in the POS tag, a number between 0 and 1, representing the probability mass assigned to the best guess tag in the tag distribution
- Named entity type, identifying person (PER), organization (ORG), location (LOC), product (PRO), event (EVE), and miscellaneous (MISC), using a BIO (or IOB2) encoding
- Base (non-embedded) phrase chunk in BIO encoding
- Token number of head word in dependency graph (according to CSI-DP)
- Type of dependency relation with head word



```
1       Marie   Marie   [Marie] SPEC(deeleigen)                         1.000000        B-PER   B-NP    2       su
2       vroeg   vragen  [vraag] WW(pv,verl,ev)                          0.532544        O       B-VP    0       ROOT
3       zich    zich    [zich]  VNW(refl,pron,obl,red,3,getal)          0.999740        O       B-NP    2       se
4       af      af      [af]    VZ(fin)                                 0.996853        O       O       2       svp
5       of      of      [of]    VG(onder)                               0.733333        O       B-SBAR  4       vc
6       hij     hij     [hij]   VNW(pers,pron,nomin,vol,3,ev,masc)      0.999659        O       B-NP    8       su
7       nog     nog     [nog]   BW()                                    0.999930        O       B-ADVP  8       mod
8       zou     zullen  [zal]   WW(pv,verl,ev)                          0.999947        O       B-VP    5       body
9       komen   komen   [kom][en]       WW(inf,vrij,zonder)             0.861549        O       I-VP    8       vc
10      .       .       [.]     LET()                                   0.999956        O       O       9       punct
```

Figure 6.1: Example output of Frog

aan, als, bij, dat, de, den, der, des, deze, die, dit, door, een, èèn, en, én, enige, enkele, er, haar, heeft, het, hét, hierin, hoe, hun, ik, in, inzake, is, jaar, je, kunnen, meer, met, moeten, na, naar, nabij, niet, nieuwe, nu, nú, of, óf, om, onder, ons, onze, ook, op, over, pas, te, tegen, ten, ter, tot, u, uw, uit, van, vanaf, vol, voor, wat, wie, wij, worden, wordt, zal, zich, zij, zijn, zullen

Table 6.1: Dutch stop words

6.2 Word weights

The importance of a word per document is determined by its weight. The weight of a word can be calculated, among others, by TF-IDF or CHI (χ^2). Other word weighting algorithms can be Information Gain and Gain Ration [9, 28]. In the subsections only TF-IDF and χ^2 are described. Each algorithm uses a word w in a document d compared to documents D, where $d \in D$. From now on the weight of a word w for a document d is referred to as: $weight(w, d)$. A document can, depending on the classifier, be:

- A sentence after applying stemming and removing stop words (the classifiers described in Section 7.2 and Section 7.4 uses this).
- A year. Each sentence comes from a specific year. After applying stemming and removing stop words these sentences are put into one document with the corresponding year.
- A category. Each sentences has none, one or multiple sentences. After applying stemming and removing stop words these sentences are put into one document (bag of words) with the corresponding category. A sentence with multiple categories goes into multiple documents, respectively (the classifier described in Section 7.3 uses this).

Every word weight is used in combination of the weight with its Part of Speech as described in Section 6.2.3. After calculating the word weights the weights are normalized per document.

6.2.1 TF-IDF

TF-IDF [23, 34, 35] stands for Term Frequency-Inverse Document Frequency. The TF-IDF weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the TF-IDF weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

TF-IDF can be successfully used for stop-words filtering in various subject fields including text summarization and classification. Typically, the TF-IDF weight is composed by two terms: the first computes the normalized Term Frequency (TF), i.e., the number of times a word appears in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears. See below for a simple example.

Each word w in a document is weighted using TF-IDF. The amount that word w occurs in document d is $\mathit{freq}(t, d)$ and $\mathit{tot}(d)$ is the total amount of words in document d. The total amount of documents is m and n is the total amount of documents the word w occurs in. The TF-IDF equation used is described in Equation 6.1:

$$\begin{aligned} \mathit{tf}(w, d) &= \mathit{freq}(w, d)/\mathit{tot}(d) \\ \mathit{idf}(w, d) &= \log(m/n) \\ \mathit{tfidf}(w, d) &= tf(w, d) \times \mathit{idf}(w, d) \end{aligned} \tag{6.1}$$

For example, consider a document containing 100 words, while the word cat appears 3 times. The term frequency (TF) for a category is then 3 / 100 $\approx$ 0.03. Now, assume we have 10 million documents and the word cat appears in one thousand of these. Then, the inverse document frequency (i.e., idf) is calculated as log(10,000,000 / 1,000) = 4. Thus, the TF-IDF weight is the product of these quantities: $0.03 \times 4 = 0.12$.

6.2.2 CHI

An alternative word weighting algorithm is the 2×2 χ^2 [14]. It calculates whether a word w is statistically significant for a document d or not. We define $occ(w, d)$ as how often a word w occurs in document d, $occ(w, \neg d)$ as how often word w occurs in other documents than document d, $occ(\neg w, d)$ as how often other words than word w occur in document d and $occ(\neg w, \neg d)$ as how often other words than word w occur in all the documents except document d. The equation is shown in Equation 6.2:

$$\chi^2(t, d) = \frac{(A + B + C + D) \times (AD - BC)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)} \tag{6.2}$$

The meaning of the letters in defined in Table 6.2. A high value means that word w is statistically significant to document d.

$A = occ(w, d)$	$C = occ(\neg w, d)$
$B = occ(w, \neg d)$	$D = occ(\neg w, \neg d)$

Table 6.2: Combing the letters in the χ^2 equation with the documented functions

6.2.3 Part of Speech

Each word w in a document is weighted in combination with Part of Speech (PoS) values per word $PoS(w)$. The part of speech tag of every word is retrieved via Frog. Only the nouns, verbs and adjunctives have a PoS value as recommended in [6, 16]. The only exception to this recommendation is the PoS "SPEC", referring to dates, numbers, names, countries, etc. ("maart", "zes", "6", "DNB", "Duitsland", etc'.). However, for instance the word "DNB" is relevant for category 1 (Macroeconomics) and names of countries are relevant for category 19 (International Affairs and Foreign Aid). The values of every PoS tag is described in Table 6.3. The equation is as follows:

$$weightPoS(w, d) = weight(w, d) \times PoS(w) \tag{6.3}$$

And $weight(w, d)$ is the word weight calculated either by TF-IDF ($tfidf(w, d)$) or CHI ($\chi^2(t, d)$).

PoS	Noun	Verb	Adjunctive	SPEC	Other
Value	1	1	1	1	0

Table 6.3: Part of Speech values

6.2.4 Normalization

There are two types of normalization used for the classifiers: L1 and L2.

L1 normalization

First way of normalization used is the normalization of a value by the sum of all the values. This kind of normalization makes, for example, more sense when working with percentages than the L2 normalization as described below. We have a value x and a series of values X, where $x \in X$, we then define:

$$L_1(x) = x / \sum_{y \in X} y \tag{6.4}$$

L2 normalization

The mostly used normalization algorithm for this research is the L2 normalization. We have a value x and a series of values X, where $x \in X$. We then define:

$$L_2(x) = \frac{x}{\sqrt{\sum_{y \in X} y^2}} \tag{6.5}$$

One of the advantages of L2 normalization in comparison with the L1 normalization is that the L2 normalization creates a bigger distance between the lowest and highest results.

Chapter 7

Classifiers

For this research four classifiers have been used: a Naive Bayes, a k Nearest Neighbor, PARC and a Support Vector Machines (SVM) classifier. For the SVM classifier some preliminary experiments are described.

7.1 Naive Bayes

This classifier is based on the Naive Bayes classifier algorithm as described in [25, 42]. It has been adjusted to handle multi-label classifications and has been optimized for the data set. It is the only classifier that doesn't work with word weights. For every sentence S that comes in the score per category C is calculated ($score(S, C)$).

Every text in the data set is put into a bag corresponding to the category. For our data set that makes 20 bags. The input sentence S is split into words. For every word w the probability per category $P(w|C)$ is calculated. The probability for each word per category $P(w|C)$ is calculated using the total number of texts in category C that contains word w: $tot(w, C)$ and the total number of texts in category C: $tot(C)$ as shown in Equation 7.1. After that the probability per category C is calculated ($P(C)$). Using the total number of texts in category C: $tot(C)$, divided by the total number of texts $tot(S)$ in the data set as shown in Equation 7.2. So we define:

$$P(w|C) = tot(w, C)/tot(C) \tag{7.1}$$

$$P(C) = tot(C)/tot(S) \tag{7.2}$$

Finally using the $P(w|C)$ for each word in sentence S, containing the words $w_1, ..., w_n$ per category and the $P(C)$ per category to calculate the score per sentence S per category C: $score(S, C)$ as follows:

$$score(S, C) = P(C) \prod_{i=1} P(w_i|C) \tag{7.3}$$

However if one word w in Equation 7.3 has a probability $P(w|C)$ of 0, then the score $score(S, C)$ is 0. To avoid this smoothing is applied. Smoothing [5] is used to avoid multiplying by 0. Smoothing also uses the total number of word, in sentence S: $tot(w, S)$, changing the $P(w|C)$ Equation to:

$$P(w|C) = \frac{tot(w, C) + 1}{tot(C) + tot(w, S)} \tag{7.4}$$

The scores for every category is then normalized using the L1 normalization as described in Section 6.2.4. The L1 normalization makes sense for the Naive Bayes classifier, because probabilities are involved.

7.1.1 Feature selection

Not all words in the sentence S are used. For every word w the $P(w|C)$ is calculated. But only the top ν words that have the highest probabilities in the sentence S are used in Equation 7.3. Here ν has been optimized from 1 to 30 with an increment of 1 (1, 2, 3, 4, 5, ..., 30). Using only where ν gives the highest score. The best result is $\nu = 4$.

7.2 *k* Nearest Neighbors

This classifier is a k Nearest Neighbor (k-NN) algorithm based on the algorithm described in [41]. It has been adjusted to handle multi-classifications and has been optimized for the data set. It looks at whether the k training texts are most similar to the input sentence S. The word weights are calculated per sentence.

We now make $m = 9143$ texts $T_0 \ldots, T_{9143}$; each text T; is a labeled sentence. Of text T the stop words, shown in Table 6.1, are removed and stemming is applied using Frog [28]. The Part of Speech is also retrieved using Frog. Each word w in text T is weighted using TF-IDF [23] in combination with part of speech (PoS) values per word. After the words are weighted the weights are normalized using the L2 normalization per sentence, as described in Section 6.2 .

After preparing the texts we are ready for the classification. When a new sentence S comes in the stop words are removed, stemming is applied, PoS information is retrieved, the words are weighted and the weights are normalized. For every sentence S the similarity is calculated per text T. A vector is created based on the word weights of text T and sentence S as $vec(T)$ and $vec(S)$. Than using cosine similarity the similarity is calculated using the following equation:

$$sim(T, S) = \frac{(vec(T), vec(S))}{\sqrt{(vec(T), vec(T))} \times \sqrt{(vec(S), vec(S))}} \tag{7.5}$$

Here the standard inner product is (-,-) used. The similarity scores of all the 9143 texts are ordered by descending similarity and the top k similarity texts are used for the prediction. Finding a value for k is documented in Section 7.2.1. The value a k has been optimized with of value of 20.

All the categories that occur in the top k similarities are used for the prediction. For example we have the results in Table 7.1 for sentence S with $k = 3$. The top scores contain the categories 1, 3 and 15. Category 1 has a summed value of 1.1, category 3 has a summed value of 0.5 and category 15 has a summed up value of 0.5. After normalizing, using the L2 normalization, the values for category 1 is: 0.84; for category 3 it is: 0.38 and for category 15 it is: 0.38. Sentence S has been classified as category 1 with a score of 0.84, as category 3 with a score of 0.38 and as category 15 with a score of 0.38.

Document	category	Similarity
s_{300}	1	0.8
s_{50}	3, 15	0.5
s_{125}	1	0.3

Table 7.1: Example values with $k = 3$ using k-NN algorithm

7.2.1 Finding k

To measure the accuracy of the algorithm one Speech from the Throne q is left out of the data set and the other ones are converted to documents. For q the scores are calculated per sentence per document. The scores are than vectorized in order of the category ids (p). The actual answers are also vectorized in order of the category ids with a value of 1 per category (a). The similarity is measured between the two vectors using the cosine similarity measurement as follows:

$$sim(p, a) = \frac{p \cdot a}{\sqrt{p \cdot p} \times \sqrt{a \cdot a}} \tag{7.6}$$

Here $p \cdot a$ denotes the standard inner product of p and a.

An average of all the sentences is calculated over all the sentences in q. The above process has been repeated for $k \in \{1, \ldots, 185\}$, with an increment of 1 (1, 2, 3, ..., 185). The best total average over all the tested Speeches from the Throne had a value of 20 for k.

7.3 Political Attention Radar Classifier (PARC)

PARC stands for Political Attention Radar Classifier. For every sentence S that comes in the score per category C is calculated ($score(S, C)$). Every text in the data set is put into a bag C corresponding to the category. For our data set that makes 20 bags. For every unique word u in the bag C the word weight is calculated using the word weight algorithms described in Section 6.2 and normalized per bag using the L2 normalization $L_2(u, C)$. For sentence S the stop words are removed and stemming is applied. Then the score is calculated per category $score(S, C)$ as follows:

$$score(S, C) = \frac{1}{n} \times \sum_{i=1}^{n} L_2(w_i, C) \tag{7.7}$$

Where w is a word in sentence S, n is the total amount of words in sentence S and C the category. If a word does not exist in all the categories the default score for that word is 0. The scores are ranked from highest to lowest. This classifier is referenced as “PARC-1”.

7.3.1 Minimum Score

All 20 categories have an equal chance. To change this a minimum score ν has been added. Only the scores that are equal to or higher than the minimum score ν are returned as predictions. Optimized ν from 0.0 to 1.0 with an increment of 0.1 (0.1, 0.2, 0.3, ..., 1.0). Removing all the predictions with less than the minimum score in the top k results. The best result is for the current data set turned out to be $\nu = 0.3$.

7.3.2 Using keywords (PARC-2)

This sections describes an algorithm (PARC-2) to find keywords that are a representation of a category. For the explanation it is better if I just start with an example.

It start by removing stop words, apply stemming to the corpus and calculate the TF-IDF weight for each word per category. An example is shown in Table 7.2.

	w_1	w_2	w_3	w_4
C_1	0.4	0.1	0	0
C_2	0.2	0.3	0.25	0
C_3	0	0	0.1	0.2

Table 7.2: Example TF-IDF values per word w per category C

We then take the sentence: "w_1 w_2 w_3 w_4" that has been labeled as category C_2. We take the top scoring word for category C_2 which is word w_2 with a score of 0.3. This word will undergo 2 checks. Check 1: if the weight in category C_2 is the highest for this word of all categories. Which is the case. Than we check (check 2) for every other word if in every other category if the word w_2 scores higher. Which is not the case, word w_1 scores higher in category C_1. It is important to check this. Chances are that word w_1 will be a keyword for category w_1, because of its high weight. It failed one of the 2 checks, thus keyword w_2 will not be marked as a keyword.

Next we take the top 2 scoring words of category C_2. Which are: w_2 and w_3. We check if the combined weight (weight is: 0.55) of these words in category C_2 is the highest in the other categories. For category C_1 this is: 0.1 and for category C_3 this is 0.1. So this check is passed. Than we check if top 2 scoring words in the other categories score less than the score of the candidate keywords. For category C_1 this is: 0.5 and for category C_3 this is 0.3. This means the words w_2 and w_3 are a good representation for category C_2 for the sentence, so these keywords will be added to category C_2. If the 2 word didn't passed the checks than the top 3 words will be picked. After that the top 4, etc. If all the words in the sentence combined is not a good combination than the sentence is ignored. It is a sign that the sentence is not a good representation for that category.

For category C_2 we have identified several keywords as shown in Table 7.3. The sentence "$w_4\, w_3\, w_5\, w_8\, w_7\, w_9$" comes in the program for classification. The classification is done per category. Only the identified keywords attached to a category may be used for classification. Other words will be ignored. For the sentence "$w_4\, w_3\, w_5\, w_8\, w_7\, w_9$" for category C_2 this means only the words "w_5, w_4, w_9" will be used. Keyword w_6 isn't in the sentence and the combination w_1 with w_3 isn't in the sentence.

w_1, w_3
w_6
w_5, w_4, w_9

Table 7.3: Possible keywords for category C_2

If each category scores 0, than the prediction is category 0 (no category) with a score of 1. If in the predictions there are 2 or more categories with a score higher than 0 and one of these predictions contain category 0. Than the prediction with category 0 is removed.

7.3.3 Improvement (PARC-3)

During the project the output of this classifier has been reviewed a couple of times in order to improve the results. In order to improve the results is to remove predictions that have a average word weight score less than γ. This means that if the average word weight score for a category is lower than γ than that category is removed from the predictions. Unless one or more words scored higher or equal than λ. The value 0.01 for γ and 0.1 for λ seems better on intuition. The use of γ and λ is not used in combination with the keywords as described in Section 7.3.2. If no predictions remain, than the prediction is category 0 with a score of 1. Basically the prediction is "unknown". These adjustments creates a new classifier: PARC-3. An overview of the these adjustments is thast the predictions of PARC-1, is only a prediction if it is conforms to the following rules:

1. Minimum score of ν (in our case it is: 0.3) after the L2 normalization.
2. Have an average TF-IDF weight of γ (in our case it is: 0.01) or higher.
3. Unless one word in the prediction scores λ (in our case it is: 0.1) or higher.

7.4 Support Vector Machines

Support vector machines (SVMs) were introduced in [29] and the technique basically attempts to find the best possible surface to separate positive and negative training samples. The “best possible” refers to the surface which produces the greatest possible margin among the boundary points.

SVMs were developed for topic classification in [7]. The paper motivates the use of SVMs using the characteristics of the topic classification problem: a high dimensional input space, few irrelevant features, sparse document representation, and that most text categorization problems are linearly separable. All of these factors are conducive to using SVMs because SVMs can train well under these conditions. That work performs feature selection with an information gain criterion and weights word features with a type of inverse document frequency. Various polynomial and RBF kernels are investigated,

C	σ	Accuracy	Description
0.5	4	0.1695	The best result from trying C from -5 to 12 and σ from -12 to 5.
0.8	0.000030518125	0.1434	Random value for C and σ
1	0.05	0.2089	Random value for C and σ

Table 7.4: SVM using a Radial Basis Function kernel values and accuracy

but most perform at a comparable level to (and sometimes worse than) the simple linear kernel.

The experiments are preliminary experiments. For the experiments the LibSVM[1][2][3][4] [4, 15] package is used. For the experiments the Radial Basis Function (RBF) kernel is used, because it gives good results overall [4]. The usual removing of the stop words and stemming of the words using Frog is applied for all the texts within the data set. After that vectors per text using TF-IDF values after the L2 normalization per text. The RBF kernel expects two parameters as input: C and σ. The right value of these parameters needs to be decided by trial and error. Recommended ranges, according to the papers, differ from $[-8, -8]$ to $[0, 001, 1000]$. See Table 7.4 for the results of some experiments. The accuracy is measured using the built in Cross Validation. The original sample is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining $k1$ subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged (or otherwise combined) to produce a single estimation. For this preliminary experiment k has been set to 5. To improve the results a better C and σ need to be calculated by trial and error.

[1] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

[2] `https://github.com/cjlin1/libsvm`

[3] `https://github.com/nicolaspanel/libsvm.net`

[4] `http://ntu.csie.org/~piaip/svm/svm_tutorial.html`

Chapter 8

Results

To measure the accuracy of the algorithm one Speech from the Throne q is left out of the data set and the rest are converted to documents. For q the scores are calculated per sentence per document. The scores are than vectorized in order of the label id's (p). The actual answers are also vectorized in order of the label id's with a value of 1 per label (a). The similarity is measured between the two vectors using the cosine similarity as follows:

$$sim(p,a) = \frac{p \cdot a}{\sqrt{p \cdot p} \times \sqrt{a \cdot a}} \tag{8.1}$$

To evaluate the results the cosine similarity is used. It as follows; we have the following categories: $\langle 1, 2, 3, 4, 5 \rangle$. For every sentence we calculate the cosine similarity. For example a sentence has been classified with categories 2 and 3, the vector for that sentence will be: $\langle 0, 1, 1, 0, 0 \rangle$. The predictions for that sentence is: $\langle 0.2, 0.1, 0.3, 0.2, 0.2 \rangle$. The cosine similarity score for that sentence will be 0.603. An overall average of the similarity of all the tested sentences is used for the results.

The Speeches from the Throne from 1945 to 2010 are all classified. To test the algorithm, the leave-one-out approach is used, one speech will be selected. All the other speeches will be used in the data set. The selected speech will be used to test the algorithm. After the algorithm is done, the outcome will be checked by the existing classification. This test will be repeated several times for the speeches 2000–2010. In the Speeches from the Throne from 2000–2010 contain 1493 sentences.

The first result is the accuracy per classifier per category and the average of every classifier. This is shown in Figure 8.1. Note that the accuracy is measured with the predictions of the classifier and not the score per category. For instance the score per category is: $\langle 0.001, 0.25, 0.3, 0.4 \rangle$, the prediction can be: $\langle 0, 0, 0.3, 0.4 \rangle$. The prediction is only used in the cosine similarity measure. The classifier that scores the best is the “PARC-3” classifier with an average of 0.675. It scores better on every category except for categories 0 and 19, where the k Nearest Neighbor classifier scores better. Note that for the “PARC-3” classifier is the only classifier which has the prediction “unknown”. When a prediction is “unknown” it not used in the accuracy. Of the 1493 sentences there are 349 sentences labeled as “unknown”.

8.1 Clean bags

The problem with a multi-label classification data set is that the same text is in different categories. The word “war” can occur in the category “agriculture” if these words have been mentioned in the same sentence, thus contaminating the category. In order the keep the categories “clean” a test is written, to only use the single classified sentences in the data set for future predictions. When running this test for the “PARC-3” classifiers it gets an average of 0.586 instead of the 0.675 it first had.

8.2 Balanced bags

The amount of sentences within each category differs: where category 19 is the biggest with 1385 sentences and category 8 the smallest with 105 sentences. For the results below, as described in [3], each category gets the same amount of sentences in each corresponding bag. Because category 8 has the smallest amount of sentences within its category, that category size determines the amount of sentences the other categories can maintain in the bag. These sentences are randomly picked. To verify the effectiveness the classifier run multiple times (20 times). When running this test for the “PARC-3” classifiers it gets an average of 0.504 instead of the 0.675 it first had.

8.3 Without exclusion of the year

Normally when classifying a year we use the leave-one-out approach. However, we wanted to see how good the classifiers are performing when the actual year are in the bags. The accuracy is shown in Table 8.1. The average for the “PARC-3” classifier increases from 0.675 to 0.711.

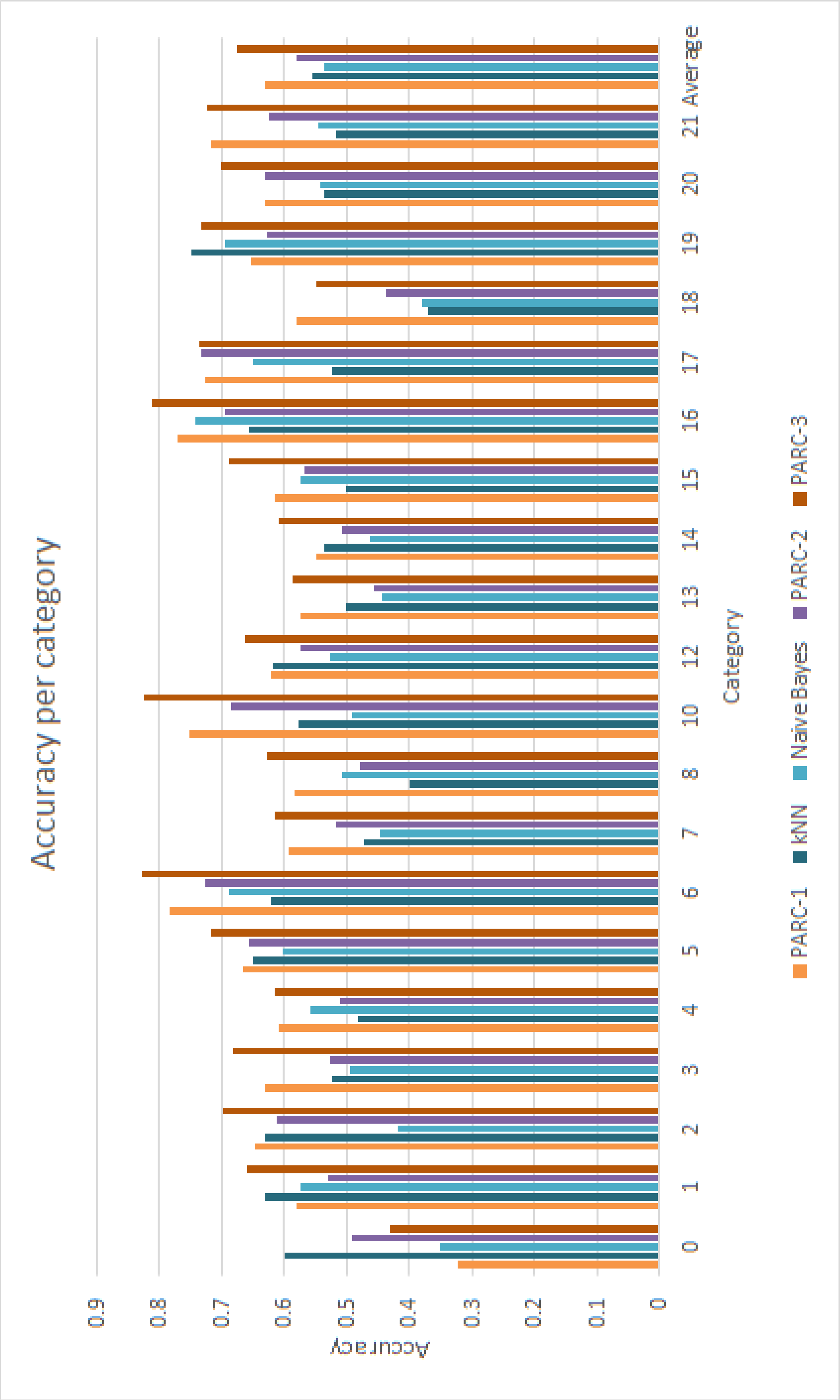


Figure 8.1: Score per category and total average

Year	PARC-3
2000	0.712
2001	0.730
2002	0.668
2003	0.752
2004	0.731
2005	0.791
2006	0.759
2007	0.697
2008	0.719
2009	0.649
2010	0.610
Average	0.711

Table 8.1: Results without excluding the classifying year

8.4 Top scoring words per category and year

To get some insight into the categories and years, the idea came up to get the top 50 scoring words per category and the top 20 scoring words per year. The top 50 scoring words per category are shown in Appendix A, and the top 20 scoring words per year are shown in Appendix B.

Also the top scoring sentences for the whole time period and the top scoring sentences per year are documented. The score is calculated by taking the sum of all the word weights in the sentence divided by the total amount of words in the sentence. The top scoring sentences of all time are documented in Appendix C. And the top 5 scoring sentences per year are documented in Appendix D.

8.5 Output examples

In this section some of examples of the predictions with the sentences are described. It also describes what can go wrong and what goes right. The output is the predictions that come from the “PARC-1” classifier. However the problems also occur in the “PARC-3” classifier.

The first sentence is “De waterkwaliteit de opslag van water met het oog op het peil van onze grote rivieren en de bodemdaling in het westen van ons land vragen veel aandacht”. This sentence is categorized as ”Public Lands and watermanagement” and ”Environment”. The prediction is the same. The words ”waterkwaliteit” and ”water” where decisive for the correct prediction.

The second sentence is “In nauwe aansluiting bij de afspraken die hierover eerder dit jaar binnen de Europese Unie zijn gemaakt zullen extra uitgaven voor onderwijs onderzoek en technologische vernieuwing worden gedaan”. This sentence is categorized as “Education” and ”Science & Technology”. The prediction has one extra category and that is category “International Affairs & Aid”. Which is arguably a correct prediction, because the EU is mentioned in the sentence. When measuring the accuracy for this sentence it not 100% correct, because ”International Affairs & Aid” is not listed as a category for that sentence.

Sometimes a lucky prediction can happen. For example the sentence “Vanuit de voorspoed van nu werken wij aan de welvaart van morgen”. It is categorized as “Macroeconomics”. The predictions are several categories including the category “Marcoeconomics”. Such lucky predictions usually happen when the word weights for every word in the sentence score low for all the categories. The “PARC-3” has the γ and λ parameters, as described in Section 7.3.3, in order to avoid low scoring categories.

Sometimes a predictions is incorrect because it is a reference to the previous sentence. For instance take the sentence “Internationaal zijn al stappen gezet ter verscherping van het toezicht”. This sentence is categorized as “Backing & Commerce”. This category does not appear in the predictions. The sentence on its own doesn’t give away that is should be that category. It is a reference to the sentence “Een sterke en stabiele financiële sector is voor onze samenleving en ons internationaal opererende bedrijfsleven van bijzonder belang”, which is also categorized as “Backing & Commerce”. For this sentence the prediction does include “Backing & Commerce”.

Another example when it can go wrong is when a new word occurs. Consider the sentence “Maar maatregelen zijn noodzakelijk omdat de stijgende levensverwachting de pensioenuitkeringen onder druk zet”. This sentence is categorized as “Labor & Employment”. The word “pensioenuitkeringen” should be the keyword to determine the prediction. However, this is a new word. This means it has no word weight and therefore cannot create a correct prediction for that sentence. The word “pensioenuitkeringen” contains the word “pensioen”, which is known. The classifier would have made a correct prediction if it would have stated “pensioen” instead of “pensioenuitkeringen”.

The sentence “Aan het begin van deze eeuw beleeft ons land een periode van economische voorspoed” has been categorized as “Macroeconomics”. The prediction is “Social Welfare”. The word that gives the right prediction should be “economische”. This word occurs 33% of the the total occurences in all the categories in the category “Macroeconomics”. However it occurs in every category and when using the IDF equation in the TF-IDF word weight equation, if a word occurs in all categories it gets a weight of 0.

8.6 First look at ANP data set

The ANP data set is that contains only Dutch news articles from the years 1993–2006. There has been some tests to use the classifier to classify a different data set. This data set contains Dutch news articles. The challenges is that it needs to classify texts based on multiple sentences instead of one sentence. The data goes from 1993 to halfway 2006. Every news article contains the following: a date, source, title, number, text content and sometimes the subjects (sport, politics, movie, etc.). All sources seem to be from “Algemeen Dagblad” (a newspaper from the Netherlands). Not every news article contains the subjects. The news articles from the years 2004 to 2006 contain no subjects. An example of a news article as-is in the data set can be found in Table 8.2.

January 19—Algemeen Dagblad—Karadzic: Geen bezwaar meer tegen aflossen VN-soldaten—1994—134— GENEVE - Een Nederlands bataljon kan medio februari Canadese soldaten aflossen in de oostelijke moslim-enclave Srebrenica. Dit zei gisteren de leider van de Bosnische Serviers, Karadzic, die geen bezwaar meer heeft tegen de aflossing. Hij nam deel aan het vredesoverleg over Bosnie in Geneve, dat werd hervat. De concessie van Karadzic, die de wisseling lange tijd heeft geblokkeerd, lijkt bedoeld om mogelijke luchtaanvallen van de Navo af te wenden. De Navo had vorige week haar bereidheid om militair in te grijpen beklemtoond. De VN willen de Canadezen in Srebrenica ontzetten en een heropening van de luchthaven van Tuzla, zodat hulpvluchten weer mogelijk worden. Reuter Krijgsmacht ; Vrede en Veiligheid ; Defensie ; Vredesmachten ; Recht ; Politiek en Staat ; Volkenrecht

Table 8.2: One ANP news article

Table 8.3 shows information per year. The data set also contains incomplete news or just some headlines, see for an example in Table 8.4. These data sets can be filtered on subject, for example “Defensie”, “Politiek en Staat”, “Economie”, “Krijgsmacht”, etc., and be classified on either title, content and/or subjects. The text needs to be cleaned from HTML codes.

We selected 25 ANP news articles from the year 2000 to 2006 at random to classify with the classifier described in Section 7.3.3 (PARC-3). Some articles were picked that contained “politiek” or “defensie” and the rest where randomly picked news articles. The articles were classified and counted how often all the categories are in the predictions. The outcome can be:

1. Category 0, count 9
2. Category 12, count 3

Year	Size	Count
1993	2945 KB	1346
1994	2962 KB	1247
1995	3166 KB	1244
1996	3013 KB	1140
1997	3603 KB	1543
1998	5071 KB	2006
1999	5439 KB	2046
2000	5211 KB	2101
2001	5152 KB	2214
2002	4558 KB	2731
2003	3673 KB	2842
2004	3245 KB	2780
2005	2654 KB	2172
2006	664 KB	673

Table 8.3: Information regarding the ANP data set per year

December 30—Algemeen Dagblad—Wie denkt Sam Burrows wel dat-ie is?—1998—14— Foto: Op pagina 23: Een jochie dat met een been in de Premier League staat

Table 8.4: One incomplete ANP news article

3. Category 1, count 1

We propose to classify a news article with a category if 3 or more sentences are classified with a category. And must not differ 9 or more from category 0. To avoid big news articles with an outcome such as:

1. Category 0, count 33
2. Category 16, count 5
3. Category 1, count 3
4. Category 8, count 2
5. Category 7, count 2
6. Category 5, count 1

For example we have the news article as shown in Table 8.5. When classifying every sentence with the "PARC-3" classifier the count of the categories is; category 12 has 4 sentences, category 0 has 1 sentence and category 20 has 1 sentence. For this news article it is save to say that the prediction is category 12. Another approach is to use the TF-IDF word weight to get the top 5 (variable) highest scoring words based on their word weight. And only use these word to classify the entire news article.

Politie moet meer bekeuren ; BINNENLAND. De politie moet de komende vier jaar 180.000 extra bekeuringen uitschrijven. Dat staat in het prestatiecontract dat het kabinet met de politie wil afsluiten. Ook moeten zaken van jeugdige criminelen veel sneller worden afgehandeld. 3 Politie ; Politiek en Staat ; Openbaar Bestuur

Table 8.5: Another example of a ANP news article

Chapter 9

Conclusion and future research

This thesis presents several classifiers for multi-label classification. It continues on the data from [3] as described in Section 3.2. It also provides new insight for future research.

TF-IDF is heavily used. It has a drawback when using multi-label classification. For instance if a word occurs in all the documents the word gets a word weight of 0. There aren't very many categories, only 20. Especially when many labels occur there is a big chance that a word is in all the categories. This can be troublesome for some words. Take, for example, the word "economische". This word occurs for 33% in the category "Macroeconomics". However it occurs in every category. This word is important for the category "Macroeconomics", but cannot be used in combination with the TF-IDF word weight algorithm. This doesn't mean that the word weight algorithm isn't good. For the majority of the words this word weight algorithm works as it should, just look at the top scoring words per category in Chapter A.

The "PARC-2" classifier does not have a high accuracy. The identified keywords per category do represent the categories very well. However the accuracy isn't nearly as good as that of the "PARC-1" classifier. This uses a minimum score, as a measure in order to ensure that not all of the predictions are returned, which is a fix to solve the problem. When only using keywords this problem doesn't occur. Only the categories that are being hit by the keywords are returned. Also the k Nearest Neighbor classifier only gives back the categories that occur in the top 20 results. This means no thought need to be put in changing the output. The "PARC-2" classifier will probably work the best on the ANP data set as it is right now, because the result really can be 0 instead of a low score of all the categories. But it will probably work the best and the same goes for the other classifiers (k-NN, PARC-1, PARC-3 and Naive Bayes), on small texts such as sentences and cleaned tweets (remove punctuation marks, remove hash tags such that only the words are left).

It is possible to rise the minimum score for the "PARC" based classifiers. The minimum score at this moment is 0.3. On the scale from 0 to 1.0, the minimum score of 0.3 seems low. When raised the classifiers will return less results. But when it comes to the accuracy of the classifiers the minimum score of 0.3 scores the best. An opportunity is to also increase the amount of "don't know" predictions for the "PARC-3" classifier.

There is still a problem with combined words such as "defensieorganisatie", "pensioenuitkeringen" and "scholingsprogramma". All Germanic based languages[1][2][3] have this problem. For the classifiers it would be better, in case of these combined words, that original words get identified and used instead of the combined words. For the combined word "defensieorganisatie" the classifier would work better with just the word "defensie". The same goes for "pensioenuitkeringen" with "pensioen" and "scholingsprogramma" with "scholing". The classifiers could perform better. The language English doesn't have this problem.

Another issue is that words like "regering", "economische", "onderwijs", "ontwikkeling", "nieuw", "jaar", "land" and "sociale" occur in every year. However, for example, the word "economische" occurs in every category, but occurs 33% of the time in category 1. That seems important for category 1, but since it occurs in every category it has a weight of 0.

The classifiers have not been tested on different kinds of data sets apart from the Speeches from the Throne. The classifiers work fine with small texts, such as sentences and tweets. However when using the classifiers to classify medium sized or bigger documents the classifiers will not work properly. Some research with feature selection needs to be done in order to make the classifiers work with medium sized or bigger documents. Other Dutch data sets are, besides the ANP data set, be:

- "de Volkskrant on CD ROM, 1997" used in [10];
- Dutch law data set as used in [8].

A lot of sentence is are a reference to an earlier sentence, basically further explaining an earlier sentence. For instance take the two sentences: "Less new recruits will be recruited this year for the army. Even less than previous year.". The first sentence is clearly about the army (category 16 Defense). The second sentence is a reference to the previous sentence, but the word army doesn't appear in it. So the classifier doesn't know if it is about the army.

[1] https://en.wikipedia.org/wiki/Germanic_languages

[2] https://nl.wikipedia.org/wiki/Germaanse_talen

[3] https://nl.wikipedia.org/wiki/Indo-Europese_talen

Some sentences that have an incorrect prediction or don't have a prediction (category 0) are enclosed within sentences that do have a prediction within the same paragraph. When three sentences $\langle s1, s2, s3 \rangle$ are categorized as $\langle 3, 0, 3 \rangle$, then probably the middle sentence should also belong to category 3. These sentences are often short, are reference to the previous line and/or add extra information to the context. For example, such a sentence can be "This is a good thing".

Looking back at the research goal we see that the goal is an automated multi-label classifier. The big question is if this goal is achieved. The answer could be both negative and positive. The "PARC-3" classifier is, unlike humans, a deterministic classifier. A person can classify a sentence with category 1 and the next year that same sentence with category 5. This classifier does not. The classifier is also good for fast classifications. However with a accuracy of $\approx 67\%$, the accuracy isn't high enough to fully rely on the predictions. The predictions still need to be verified by humans. The classifier can also be used to classify the Speeches from the Throne and then the sentences with the predictions by humans, to make it less cumbersome for the human coders. Concluding, the goal is achieved for fast results, however the accuracy is not high enough to fully rely on the predictions.

9.1 Final product

The "PARC-3" classifier is put into a website[4] as the final product. The classifiers works just as described as in this document. A screen-shot of the website presented in Figure 9.1.

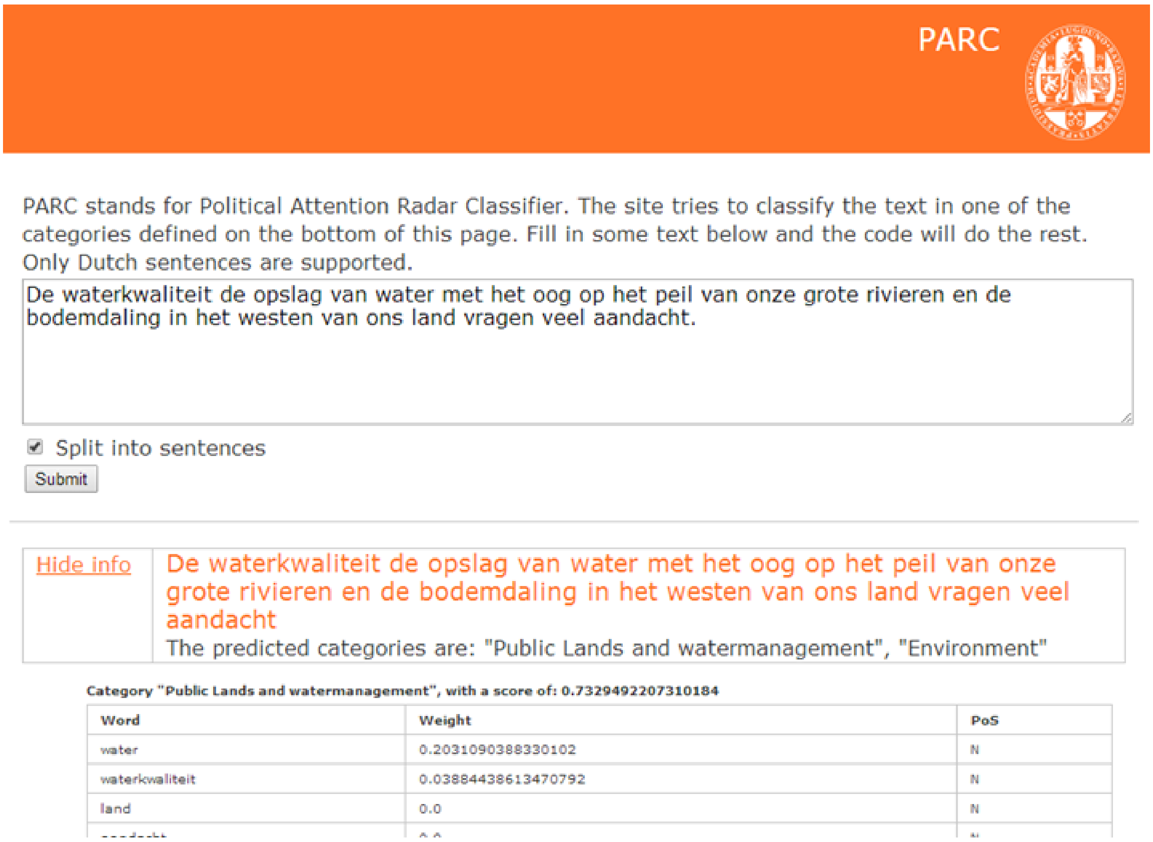
If an user inputs some text and clicks on "Submit" the website shows the text, o.a., as in Figure 9.2. If you mouse over a sentence you will get more details about the classification of that sentence.

9.2 Future research

Still some points stand open in order to make the classifiers better. One of these points is the problem with combined words. Still no real answer has been found for this problem.

Another point is to use information within a paragraph in order to identify the context of the paragraph. This information can be used to better classify the sentence. Even sentences that have been categorized with "unknown" can be categorized using the information of the context.

[4] `http://parc.deifier.eu.cloudbees.net/`



PARC stands for Political Attention Radar Classifier. The site tries to classify the text in one of the categories defined on the bottom of this page. Fill in some text below and the code will do the rest. Only Dutch sentences are supported.

De waterkwaliteit de opslag van water met het oog op het peil van onze grote rivieren en de bodemdaling in het westen van ons land vragen veel aandacht.

☑ Split into sentences

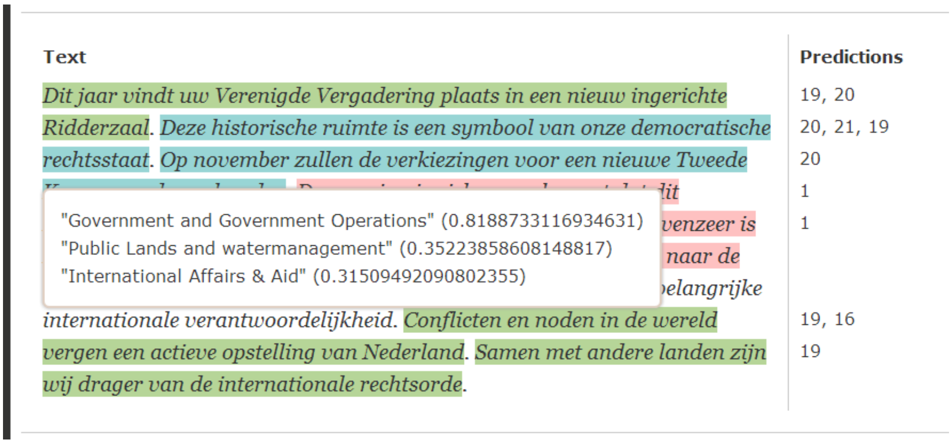
De waterkwaliteit de opslag van water met het oog op het peil van onze grote rivieren en de bodemdaling in het westen van ons land vragen veel aandacht

The predicted categories are: "Public Lands and watermanagement", "Environment"

Category "Public Lands and watermanagement", with a score of: 0.7329492207310184

Word	Weight	PoS
water	0.2031090388330102	N
waterkwaliteit	0.03884438613470792	N
land	0.0	N
aandacht	0.0	N

Figure 9.1: Screen-shot of the classifier website



Text	Predictions
Dit jaar vindt uw Verenigde Vergadering plaats in een nieuw ingerichte Ridderzaal.	19, 20
Deze historische ruimte is een symbool van onze democratische rechtsstaat.	20, 21, 19
Op november zullen de verkiezingen voor een nieuwe Tweede	20
dit	1
venzeer is	1
naar de	
belangrijke internationale verantwoordelijkheid.	
Conflicten en noden in de wereld vergen een actieve opstelling van Nederland.	19, 16
Samen met andere landen zijn wij drager van de internationale rechtsorde.	19

"Government and Government Operations" (0.8188733116934631)
"Public Lands and watermanagement" (0.35223858608148817)
"International Affairs & Aid" (0.31509492090802355)

Figure 9.2: Output of the website

When getting feedback about the classifier, one of the first things that came up was if the classifiers could be used for texts in English. There are only two things needed to support English: an English lemmatizer and an English coded data set. The most used stemming solution for English is the Porter stemming algorithm [39]. From a computer science perspective a lemmatization that supports English words, and preferably also supports Dutch words, needs to be found.

The current data set also has sub-categories, which hasn't been mentioned before. The classifier could also be used to classify sentences in these sub-categories. There are two ways to classify the sub-categories:

1. In two phases. The first phase is to classify the main categories. The second phase is to identify the sub-category within the main categories.

2. In one phase. Try to classify the sub-categories directly and extract the main category from the sub-category.

Another target is to make changes to the classifiers so that they also support medium to big sized documents. Some research is needed with feature selection to realize this goal. When done so, the classifiers can classify news articles as in the ANP data set, "de Volkskrant on CD ROM, 1997" data set and the Dutch law data set. Another data set is the Reuters data set[5][6]. When the Reuters data set is classified the classifiers can be compared to other classifiers.

It is also possible to add the feature that the classifiers support input from users. This will enable users to make real-time corrections to predictions. In order to keep up with new words, support words that aren't in the data set, change the weight of words (words with a low occurrence have a low score, but might be important) and keep up with the changes of a word (Germany in 1945 was category defense and now it is foreign affairs).

Another way to classify texts is to use the Latent Semantic Analysis (LSA) technique to analyze the corpus. LSA can be used to analyze the relationship between a word and a category. The steps of LSA [17, 43] are as follows:

1. Make a 2d matrix, where the rows are the words and the columns the documents. The matrix is filled with the information on how often a word occurs in a document: the frequency of each word per document.

2. Calculate in the matrix the TF-IDF values of each word and document.

3. Normalize the TF-IDF values.

[5] http://archive.ics.uci.edu/ml/datasets.html

[6] http://www.daviddlewis.com/resources/testcollections/reuters21578/

4. Singular Value Decomposition.

The Part of Speech values as described in Section 6.2.3 need to be optimized. Now the PoS values for nouns, verbs, adjunctives and SPEC all have the same value of 1. This does not have to be the case. A noun can be, for example, more important than an adjunctive. These values need to be optimized by running the classifier and systematically testing different values.

The "PARC-3" classifier is, according to the results, clearly the best classifier, except for category 0. For this category the "kNN" classifier is clearly the best. To improve the result a hybrid solution can be created, using the "kNN" classifier for the category 0 predictions and the "PARC-3" classifier for the predictions of the rest of the categories.

Some overlap occurs in the categories. When, for example, the word "pensioen" occurs in a sentence that sentence can be classified with category 5 "Labor, Employment and Immigration" and category 13 "Social Welfare". These categories often occur together. These categories are probably not the only ones. Some research is needed to identify categories which often occur together. This information can be used to improve the predictions.

Bibliography

[1] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[2] Gerard Breeman, David Lowery, Caelesta Poppelaars, Sandra L. Resodihardjo, Arco Timmermans, and Jouke de Vries. Political attention in a coalition system: Analysing queen's speeches in the Netherlands 1945–2007. *Acta Politica*, 44 (1):1–1, 2009.

[3] Gerard Breeman, Hans Then, Jan Kleinnijenhuis, Wouter van Atteveldt, and Arco Timmermans. Strategies for improving semi-automated topic classification of media and parliamentary documents. In *2nd Annual Meeting of the Comparative Policy Agendas Conference, The Hague*, 2009.

[4] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[5] Stanley Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, pages 310–318, 1996.

[6] Stephanie Chua. The role of parts-of-speech in feature selection. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2008.

[7] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[8] Emile de Maat and Radboud Winkels. A next step towards automated modelling of sources of law. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 31–39. ACM, 2009.

[9] Franca Debole and Fabrizio Sebastiani. Supervised term weighting for automated text categorization. In *Text Mining and its Applications*, pages 81–97. Springer, 2004.

[10] Tanja Gaustad and Gosse Bouma. Accurate stemming of Dutch for text classification. *Language and Computers*, 45(1):104–117, 2002.

[11] Jean-Luc Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298, 1994.

[12] Nadia Ghamrawi and Andrew McCallum. Collective multi-label classification. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 195–200. ACM, 2005.

[13] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2006.

[14] Kotaro Hashimoto and Takashi Yukawa. Term weighting classification system using the chi-square statistic for the classification subtask at NTCIR-6 patent retrieval task. In *Proceedings of the 6th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access (NTCIR'07)*, pages 385–389, 2007.

[15] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A Practical Guide to Support Vector Classification. Technical report, Department of Computer Science, National Taiwan University, 2003.

[16] Chuntao Jiang, Frans Coenen, Robert Sanderson, and Michele Zito. Text classification using graph mining-based feature extraction. *Knowledge-Based Systems*, 23(4):302–308, 2010.

[17] Thomas Landauer, Peter Foltz, and Darrell Laham. An introduction to Latent Semantic Analysis. *Discourse Processes*, 25(2-3):259–284, 1998.

[18] Daniel Larose. *Discovering Knowledge in Data: An Introduction to Data Mining*. John Wiley & Sons, 2005.

[19] Tao Li and Mitsunori Ogihara. Detecting emotion in music. In *Proceedings of the Fourth International Conference on Music Information Retrieval*, volume 3, pages 239–240, 2003.

[20] Stephen Purpura and Dustin Hillard. Automated classification of congressional legislation. In *Proceedings of the 2006 International Conference on Digital Government Research*, pages 219–225, 2006.

[21] Stephen Purpura, Dustin Hillard, and Philip Howard. A comparative study of human coding and context analysis against support vector machines (SVM) to differentiate campaign emails by party and issues (working draft), 2006.

[22] John Ross Quinlan. *C4.5: Programs for Machine Learning*, volume 1. Morgan Kaufmann, 1993.

[23] Juan Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the First Instructional Conference on Machine Learning*, 2003.

[24] Joseph Lee Rodgers and W. Alan Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.

[25] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, 34(1):1–47, 2002.

[26] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

[27] Nadeem Ur-Rahman and Jennifer A. Harding. Textual data mining for industrial knowledge management and text classification: A business oriented approach. *Expert Systems with Applications*, 39(5):4729–4739, 2012.

[28] Antal van den Bosch, Bertjan Busser, Sander Canisius, and Walter Daelemans. An efficient memory-based morphosyntactic tagger and parser for Dutch. In *Computational Linguistics in the Netherlands: Selected Papers from the Seventeenth CLIN Meeting*, pages 99–114, 2007.

[29] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2000.

[30] Website. Stemming and lemmatization, 2008. URL `http://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html`.

[31] Website. Frog, 2014. URL `http://ilk.uvt.nl/frog/`.

[32] Website. Hyponymy and hypernymy, May 2014. URL `https://en.wikipedia.org/wiki/Hyponymy`.

[33] Website. Speech from the throne, 2014. URL `https://en.wikipedia.org/wiki/Speech_from_the_throne`.

[34] Website. tfidf, 2014. URL `https://en.wikipedia.org/wiki/Tf-idf`.

[35] Website. Tf-idf :: A single-page tutorial — Information retrieval and text mining, May 2014. URL `http://www.tfidf.com/`.

[36] Website. Computer-assisted qualitative data analysis software, May 2014. URL `https://en.wikipedia.org/wiki/Computer-assisted_qualitative_data_analysis_software`.

[37] Website. Qualitative research, May 2014. URL `https://en.wikipedia.org/wiki/Qualitative_research`.

[38] Zhihua Wei, Hongyun Zhang, Zhifei Zhang, Wen Li, and Duoqian Miao. A naive Bayesian multi-label classification algorithm with application to visualize text search results. *International Journal of Advanced Intelligence*, 3(2):173–188, 2011.

[39] Peter Willett. The porter stemming algorithm: Then and now. *Program: Electronic Library and Information Systems*, 40(3):219–223, 2006.

[40] Ian Witten, Geoffrey Holmes, Mark Hall, and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, third edition, 2013.

[41] Yiming Yang. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 13–22. Springer, 1994.

[42] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49. ACM, 1999.

[43] Kai Yu, Shipeng Yu, and Volker Tresp. Multi-label informed Latent Semantic Indexing. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 258–265. ACM, 2005.

[44] Min-Ling Zhang and Zhi-Hua Zhou. A k-nearest neighbor based algorithm for multi-label classification. In *Proceedings of the 2005 IEEE International Conference on Granular Computing*, volume 2, pages 718–721. IEEE, 2005.

[45] Min-Ling Zhang and Zhi-Hua Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

Acknowledgements

I would like to thank several people for making this research possible. First I especially thank Walter Kosters and Christoph Stettina for supervising the project, investing time on a weekly basis and giving expert knowledge for this research. Also I would like to thank Arco Timmersman for providing the data set, giving the research assignment and feedback from a social studies aspect of view. Another thanks for feedback from a social studies aspect of view to Ulrich Mans, Jamie Fleet and Gideon Shimshon. After that Jan Kalmeijer and Marijn Schraagen for additional technical support. Lastly I would like to thank my parents, my brother and friends for their continued support.

Appendix A

Top scoring words per category

Top 50 scoring words per category in order of score from high to low.

Category	Top 50 scoring words
(no category)	zegen, wijsheid, gods, zitting, bede, verklaren, toewensen, god, bidden, lid, hart, gewoon, generaal, mij, staat, openen, rust, werkkracht, toewijding, wens, troonrede, besef, gij, koninkrijk, rusten, volksvertegenwoordigers, herdenking, vertrouwen, verantwoordelijk, volks, bevrijden, mijn, mogen, wier, wereldoorlog, toevertrouwen, herdenken, wacht, beseffen, zegenen, erasmus, worden, offer, weten, diep, volk, taak, spreken, onzekerheid, veelomvattend
1	financieringstekort, belastingdruk, koopkracht, inkomstenbelasting, loon, besteding, tekort, staatsschuld, omzetbelasting, belastingmaatregel, inkomen, belasting, rentelast, prijsbeleid, conjunctuur, inflatie, werkgelegenheid, accijns, kapitaalmarkt, inkomst, stijgen, collectief, verlichting, tarief, procent, inflatiecorrectie, vennootschapsbelasting, matigen, stoffelijk, inflatoir, loonbelasting, begrotingstekort, matiging, overheidsfinancin, rijksbegroting, besparing, schuld, levensonderhoud, belastingtarief, industrie, industrieel, vertonen, btw, rente, benzine, monetair, last, vermogensbelasting, loonontwikkeling, stijging

2	asielzoeker, vrouw, toelatingsbeleid, emancipatie, inburgering, vreemdeling, gezinshereniging, discriminatie, asiel, tolerantie, vreemdelingenwet, minderheid, vluchteling, restrictief, migratiestromen, verblijven, migrant, vrijheid, nieuwkomer, respect, taal, migratie, verblijf, verdraagzaamheid, opvang, gelijk, misbruik, vreemd, herkomst, aanvraag, wetboek, meerderjarigheid, ongerechtvaardigd, achterstelling, ingezetenen, persoonsregistratie, emancipatiebeleid, illegaal, criterium, multicultureel, onevenwichtig, vreemdelingenbeleid, afwijzen, intolerantie, inburgeringsprogramma, migratiebeleid, inburgeringscursus, persoonlijk, stroom, procesrecht
3	medische, patint, wachtlijst, gezondheidszorg, betaalbaar, medisch, kostenbeheersing, verzekeraar, thuiszorg, zorgsector, gezondheidstoestand, werkdruk, volksgezondheid, ziektekostenverzekering, ziek, ziektekosten, chronisch, wachttijd, geneesmiddel, ethisch, iedereen, geneeskunde, euthanasie, beroepsbeoefenaar, volume, ziekenfonds, levensstijl, zorgverzekering, zorgstelsel, ziekenfondsverzekering, gezondheid, doelmatigheid, druggebruik, toegankelijk, kost, ziekenfondswezen, gezondheidsvoorziening, gezondheidsraad, dunning, basispakket, ethiek, drugbeleid, roken, verplegen, ongezond, zorgtoeslag, gezondheidssituatie, verpleeghuis, verkorting, drug
4	agrarisch, tuinbouw, landbouw, boer, visserij, landbouwprodukt, landbouwbeleid, landbouwkundig, landbouwbedrijf, tuinder, ruilverkaveling, noodvoorziening, landbouwpolitiek, veehouderij, kleinbedrijf, platteland, ondernemer, structuurbeleid, structuurnota, produktie, kostprijs, tuinbouwproduct, marktordening, pachtwet, landbouwonderwijs, zeevisserij, bedrijfsstructuur, garantiebeleid, overproduktie, productie, ijgen, innovatief, industrie, markt, teisteren, afzet, bestaanszekerheid, rationeel, landbouwuitgave, mestoverschot, mestproblematiek, gemeenschappelijk, a, succesvol, handel, bedrijfsvoering, overschot, overdracht, milieuprobleem, consument

5	werkgever, werkloosheid, werklozen, werkgelegenheid, werknemer, arbeidsmarkt, minimumloon, partner, loonvorming, stichting, scholing, werkzoekend, werkervaring, arbeidskracht, arbeidsproces, loonkost, arbeidsovereenkomst, uitkering, jong, langdurig, arbeidsvoorziening, matigen, matiging, tekort, loon, arbeidskosten, werkenden, baan, allochtoon, arbeidsongeschikt, bijscholing, arbeidsplaats, inkomen, vrouw, pensionering, pensioenwet, cao, jeugdwerkloosheid, jeugdwerkgarantieplan, herintreden, minderheid, beroepsbevolking, inflatie, pensioen, loonontwikkeling, zolang, herverdeling, collectief, deeltijdbanen, werkloos
6	onderwijs, beroepsonderwijs, leerling, kunst, leraar, school, middelbaar, leerkracht, klas, basisvorming, sport, cultuur, student, kind, basisonderwijs, onderwijsvoorziening, kleuteronderwijs, collegegeld, onderwijsbestel, leerlingenschaal, jarige, basisschool, leerplicht, studiefinanciering, jong, universiteit, wetenschappelijk, scholing, leerplichtig, docent, augustus, hogeschool, kleuter, lerarenopleiding, vervolgonderwijs, schooluitval, taal, vaardigheid, vorming, kennis, wetenschap, herstructurering, bloeien, museum, bijscholing, leerlingwezen, sportbeoefening, erfgoed, hooger, onderwijzer
7	co, stof, luchtverontreiniging, afval, uitstoot, milieubeleidsplan, emissie, schoon, milieubeleid, verontreiniging, milieuhygine, vervuiling, afvalstof, geluidshinder, bodembescherming, water, inspraak, lucht, schadelijk, bodemverontreiniging, rijnmondgebied, vaststelling, waddengebied, chemisch, rijn, milieugevaarlijk, hergebruik, milieuvriendelijk, milieuprobleem, broeikasgas, klimaatverandering, auto, milieuvervuiling, scheiden, bodem, waterkwaliteit, opslag, rivier, absoluut, erin, product, grondstof, energie, ecologisch, aanscherping, natuurgebied, luchtkwaliteit, vervoer, consumptie, uitwerken

8	kernenergie, energie, aardgasprijs, aardgas, energiebeleid, zonne, dieselaccijns, energieheffing, energiehuishouding, fossiel, brandstof, kool, olie, accijns, benzine, energiebesparing, atoomenergie, verbruik, mijnindustrie, gas, energieverbruik, aardgasbaten, energieprijs, vreedzaam, limburg, btw, aardgasvondsten, invoer, aanschaf, opbrengst, industrieel, zuinig, inkomst, energievoorziening, verdrag, prijs, mijnwezen, atoomkernwetenschap, spoediger, uitermate, reactor, pet, euratom, daarnevens, kernfysisch, hulpstoffen, energieverspilling, energienota, kweekreactor, looptijd
10	vervoer, verkeersveiligheid, bereikbaarheid, verkeers, auto, vervoersbeleid, mobiliteit, rijkswegenfonds, verbinding, verkeersslachtoffer, hogesnelheidslijn, betuwelijn, schiphol, motorrijtuigenbelasting, infrastructuur, rotterdam, wegverkeer, luchtvaart, zeehaven, personenauto, vaarweg, verkeersongeval, weggebruiker, wegenbouw, spoorweg, luchthaven, file, verkeer, onderhoud, accijns, zeescheepvaart, randstad, achterlandverbinding, mainports, slachtoffer, structuurschema, stad, waterstaat, achterland, autosnelweg, vooruitlopen, rijksweg, maximumsnelheden, waarborgsom, dodelijke, personenvervoer, fiets, wegennet, ergernis, automobiliteit
12	politie, rechterlijk, criminaliteit, rechtspraak, justitie, misdaad, gevangeniswezen, straat, politiekorps, misdrijf, overlast, geweld, sluitstuk, delinquent, crimineel, fraude, rechtshandhaving, orgaan, ministerie, gedrag, preventie, macht, justitieel, agressief, rechter, terroristisch, strafbaar, opsporingsmethode, onveilig, jeugdcriminaliteit, bestrijding, drug, effectief, leefomgeving, onveiligheid, misbruik, reclassering, rechtspleging, drughandel, jeugdig, voelen, reorganisatie, rechtsstaat, rechtsbijstand, vrijheidsstraf, vandalisme, strafrechtpleging, gevangeniscapaciteit, criminaliteitsbestrijding, voortgangsrapportage

13	uitkering, kinderbijslag, aow, kinderopvang, arbeidsongeschiktheid, zekerheid, oud, werklozen, ouderdomsvoorziening, kind, welzijnsbeleid, premie, ouder, arbeidsproces, vergrijzing, werkloosheidswet, ziektewet, arbeidsongeschiktheidsverzekering, volksverzekering, minimumloon, uitsluiting, gehandicapten, koopkracht, gezin, bejaard, samenleving, jong, bijstand, minimum, oudedagsvoorziening, vrijwilligerswerk, langdurig, arbeidsongeschikt, armenwet, weduwe, geneeskundig, jeugd, zelfstandig, gemakkelijk, pensioen, verzekering, koppeling, werkloos, oneigenlijk, cohesie, kwetsbaar, solidariteit, armoede, kinderbijslagwet, thuiszorg
14	woning, woningbouw, woningnood, bouw, huren, stad, volkshuisvesting, aanbouw, huurverhoging, stadsvernieuwing, bouwkosten, huursubsidie, bouwprogramma, woningwetwoning, woonruimte, huur, wijk, stedelijk, inkomensgroep, woningbouwcorporatie, bouwwerk, woningtekort, woningvoorraad, woningproduktie, woningbouwprogramma, subsidiebeleid, bejaardenoord, opvoeren, platteland, huisvesting, woonomgeving, gemeente, betaalbaar, huishouden, woningwetbouw, rijkssteun, woningbezit, woningcorporatie, opeenhoping, probleemwijk, bouwen, gereedkomen, woningverbetering, gebouw, doorstroming, inwonen, subsidie, nieuwbouw, rijk, materiaal
15	kleinbedrijf, onderneming, ondernemer, midden, middenstand, bedrijfslichaam, bedrijfsorganisatie, productschap, aanmoediging, ondernemerschap, zelfstandig, administratief, ondernemingsraad, overheidsbemoeiing, bedrijfsleven, publiekrechtelijk, afzet, starten, bedrijfsgenooten, liquiditeitsmoeilijkheden, ondernemingsrecht, vennootschap, fusie, consument, deregulering, winst, last, ondernemen, toerisme, vergunning, bedrijfschappen, consumptief, rendement, nijverheid, ser, bedrijfstak, industrie, visserij, agrarisch, advies, bezitsvorming, inkrimping, bedoeld, bedrijfsgebouw, bevoegdheid, tuinbouw, herstellen, innovatie, raad, vermogensbelasting

16	atlantisch, krijgsmacht, militair, navo, bondgenootschap, defensie, vrede, wapenbeheersing, nucleaire, verdediging, wapen, noord, strijdkracht, vredesoperatie, west, kernwapen, defensienota, conflict, bondgenootschappelijk, verdragsorganisatie, bewapening, afghanistan, dienstplichtig, conventioneel, oost, noordatlantisch, wapenwedloop, waarborg, westelijk, plaatsing, plichtsbetrachting, defensieorganisatie, marine, sowjet, irak, missie, bondgenoot, conferentie, bespreking, veiligheidsbeleid, genve, verdrag, koninklijk, landmacht, middellang, kruisvluchtwapen, ontwikkelingssamenwerking, transatlantisch, crisisbeheersingsoperatie, strijdmacht
17	wetenschapsbeleid, wetenschap, televisie, wetenschappelijk, computer, omroepbestel, omroep, radio, mediabeleid, informatie, kunst, elektronisch, technologie, speurwerk, mediawet, snelweg, kennisdebat, onderwijs, kennis, innovatie, informatietechnologie, technologisch, communicatie, onderzoekbestel, wetenschapsbeoefening, aardgasvondsten, kleinbedrijf, technisch, hoogwaardig, ondernemerschap, innovatieplatform, omgaan, onderzoek, welvaartsplan, radioraad, uitdragen, zendtijd, kijkgeld, televisiebestel, october, uitzending, televisiezendtijd, verschijning, pacificatiecommissie, televisienet, omroepwet, proefresultaat, antennesysteem, universitair, wetenschapsbudget
18	betalingsbalans, wereldhandel, betalingsverkeer, handelsverkeer, handels, ontwikkelingsland, vrijmaking, vrijhandelszone, uruguay, buitenland, concurrentie, uitvoer, handel, invoerrecht, handelsbetrekkingen, monetair, levering, betaalmiddel, tengevolge, ruilvoet, liberalisering, invoer, ongunstig, waarschijnlijk, prijspeil, wereldmarkt, concurrentiepositie, wereldhandelsconferentie, kapitaalverkeer, azi, integendeel, wereldhandelsorganisatie, voorwerp, betalingsbalanssaldo, tekort, munt, stijging, grondstof, unie, vertonen, globalisering, genve, continent, overschot, handelspolitiek, verloop, aanhouden, protectionisme, waken, opwaarts

19	antillen, suriname, koninkrijk, ontwikkelingsland, oost, statuut, afrika, ontwikkelingssamenwerking, natie, guinea, verenigen, west, aruba, unie, gemeenschap, verdrag, vrede, staatkundig, arm, indonesi, luxemburg, atlantisch, oosten, rechtsorde, benelux, toetreding, belgi, conflict, wereld, armoede, conferentie, lidstaat, politiek, relatie, hulpverlening, militair, democratisch, surinaamse, bezoek, eiland, monetair, europees, bespreking, parlement, duitsland, verdiepen, bilateraal, vietnam, zuid, tegenstelling
20	gemeente, decentralisatie, burgemeester, provincie, bestuur, verkiezing, rijksdienst, bestuurlijk, staatkundig, democratisch, grondwet, democratie, mijn, koninkrijk, ambtenaar, bestuursakkoorden, publiek, deregulering, rechtsstaat, gemeentelijk, rijk, bestel, huwelijk, overheidsapparaat, inspraak, referendum, grondrecht, kiesstelsel, belastinggebied, wetgeving, volksvertegenwoordiging, voorkeursrecht, staatscommissie, slagvaardig, rijksoverheid, binnenlands, kamer, mij, kieswet, rechtspraak, ombudsman, ontlenen, overheidspersoneel, vertrouwen, relatie, rechtshandhaving, informatie, ministerie, respect, functioneren
21	ruimtelijk, ordening, water, natuurbeleidsplan, landschap, oosterschelde, natuurgebied, groen, natuurterrein, bodem, waterbeheer, walcheren, waterkering, deltawet, waterhuishouding, stormvloedkering, deltawerk, landschappelijk, kust, zeespiegel, platteland, verontreiniging, lucht, openluchtrecreatie, leefomgeving, natuur, landaanwinning, drooglegging, watersnood, aankoop, waterschap, onvervangbaar, museum, hoofdlijn, provincie, woningbouw, inrichting, agrarisch, recreren, leefmilieu, verwant, inpoldering, ecologisch, formuleren, planologisch, dichtbevolkt, inwonen, historisch, structuurschema, monumentenzorg

Appendix B

Top scoring words per Year

Top 20 scoring words per year in order of score from high to low

Year	Top 20 scoring words
1995	awbz, drughandel, participatie, geneesmiddel, sanering, bevinden, specifiek, werkervaring, inspelen, bereiden, concurrentiekracht, koninkrijksverband, dynamiek, slagvaardig, criminaliteit, uitdagen, speerpunt, bijkomen, beleidsintensivering, openingstijd
1996	absoluut, alleenstaand, co, jeugdig, eeuwwisseling, druggebruik, ondersteuning, arbeidsproces, gezondmaking, geborgenheid, emissie, proces, zorgtaak, tellen, ondersteunen, kennis, inschakeling, democratisering, schadelijk, nadruk
1997	ongezond, omroep, levensstijl, leefmilieu, alcohol, basisschool, evaluatie, sport, drug, co, kennis, apart, werkdruk, amsterdam, uitsluiting, erkenning, reserveren, schaal, leer, cohesie
1998	toerusting, introductie, energiezuinige, geld, cohesie, minister, uiterlijk, formuleren, media, basisvoorziening, dichtbij, burger, justitie, weerbaarheid, burgemeester, energiebesparing, najaar, energie, slaan, computer
1999	eeuw, communicatie, voortijdig, hoogwaardig, stabiliteit, mens, norm, vooruitgang, dichtbij, infrastructuur, etnisch, investeren, modern, flexibel, balans, kwaliteit, veranderen, rechtsstaat, bijdragen, patint
2000	technologisch, voorspoed, woonomgeving, invulling, toename, gebruikmaken, inzet, bereikbaarheid, afspraak, menselijk, opsporing, computer, keuzevrijheid, hoogwaardig, betalen, continent, verankeren, vaardigheid, opdracht, plaatsen

2001	rechtvaardigheid, soort, voedselveiligheid, prins, euro, miljard, betrokkenheid, flexibiliteit, globalisering, kwaliteit, vreemdeling, huwelijk, intensief, aanslag, aanpassingsvermogen, keuzevrijheid, ondersteuning, vaak, schaal, adequaat
2002	loonkost, debat, naleven, tegemoetkomen, aflossing, meerjarig, drager, brandweer, streng, voortbouwen, combinatie, burger, aanslag, instantie, preventie, regelgeving, integratie, vaak, prestatie, ondernemen
2003	efficint, slagvaardig, randvoorwaarde, cultuuromslag, uitgangspunt, kenniseconomie, unie, daadwerkelijk, kerntaak, meedoen, fors, duurzaam, perspectief, productiviteit, burgemeester, medisch, droogte, regel, verstrekken, budgettair
2004	productiviteit, arbeidsparticipatie, innovatieplatform, hervorming, pluriform, toelating, universiteit, werknemer, hernieuwen, achtergrond, slagvaardig, agenda, toekomstgericht, arbeidsgeschiktheid, werkloosheidsregeling, prikkel, beroepsarbeid, ontslagvergoeding, verrekenen, werkloosheidsuitkering
2005	verwevenheid, verscheidenheid, mooi, merkbaar, ingreep, mens, relatie, veilig, bouwen, dynamisch, feestelijk, toon, hartverwarmende, enthousiasme, ambt, leefsituatie, stoppen, eeuwenlang, bevolkingssamenstelling, medemens
2006	lot, merken, meedoen, kwart, omgeving, innovatie, thuiszorg, ouder, mens, inspelen, huis, wachtlijst, ondernemer, ridderzaal, congo, bres, medicijn, terreurdreiging, meezoeken, europeanen
2007	pijler, beleidsprogramma, zorgverzekering, dierbaar, mens, mooi, internet, wijk, ondernemer, bereikbaar, missie, ondernemen, keer, inburgering, gezin, afspraak, starten, respect, leefomgeving, optimistisch
2008	leefomgeving, houvast, trein, begeleiden, fundament, gedrag, ontlenen, aanschaf, meedoen, buurt, zetten, rechtsstaat, mens, aantrekkelijk, centrum, burger, huis, crisis, leraar, arbeidsparticipatie
2009	recessie, heroverwegingen, euro, vastberadenheid, procent, overheidsfinancin, miljard, uitzonderlijk, bestuurder, medeoverheden, burgerschap, vergrijzen, crisis, staatsschuld, gedrag, handelen, eerlijk, faillissement, slinken, spaartegoed

2010	sint, levensverwachting, beveiligen, maarte, curaao, crisis, overheidstekort, opereren, stappen, herstellen, toegankelijkheid, innovatief, opzetten, stabiel, vanzelfsprekend, duurzaamheid, oplopen, afghanistan, stabiliteits, groeipact

Appendix C

Top sentences

The table below shows the top scoring sentences for the whole time period. The score is calculated by the sum of all the word weights divided by the amount of words in the sentence.

Top	Sentence	Year
1	Nederland werkt	2006
2	Solide oplossingen vragen tijd	2005
3	De criminaliteit daalt	2005
4	Herstel is mogelijk	1983
5	De wereldhandel hapert	1980
6	De motorrijtuigenbelasting verhoogd	2008
7	Een veiliger Nederland	2005
8	Waakzaamheid is geboden	1990
9	Rechtsbescherming is belangrijk	1987
10	Internationale samenwerking is geboden	1993
11	Leden der Staten Generaal	1948
12	Leden der Staten Generaal	1949
13	Leden der Staten Generaal	1952
14	Leden der Staten Generaal	1953
15	Leden der Staten Generaal	1954

Appendix D

Top scoring sentences per year

In the table below show the top scoring sentences per year. The score is calculated by the sum of all the word weights divided by the amount of words in the sentence.

Year	Top 5 sentences
1995	"De zorgsector levert kwalitatief goede prestaties", "Leden van de Staten Generaal", "Dat is positief", "Jongeren geven vorm aan de toekomst", "Actieve participatie is speerpunt van beleid"
1996	"Leden van de Staten Generaal", "De politie wordt verder uitgebreid", "De huidige ontwikkelingen zijn bemoedigend", "De gezondmaking van de openbare financin verloopt voorspoedig", "Een krachtige economische expansie is daartoe onontbeerlijk"
1997	"Miljoenen mensen zijn ontheemd", "Kennis moet worden bijgehouden", "Bijzondere aandacht vraagt de jeugdcriminaliteit", "De werkgelegenheid blijft krachtig doorgroeien", "Leden van de Staten Generaal"
1998	"Hiervoor komt extra geld beschikbaar", "Dit noopt tot behoedzaamheid", "Leden van de Staten Generaal", "De mobiliteit neemt snel toe", "Veiligheid gaat de gehele gemeenschap aan"
1999	"Twee bloedige wereldoorlogen werden uitgevochten", "Leden van de Staten Generaal", "Velen hebben daaraan bijgedragen", "Goed openbaar bestuur inspireert tot actief burgerschap", "Een degelijk financieel economisch beleid blijft geboden"

2000	"Leden van de Staten Generaal", "De werkgelegenheid ontwikkelt zich gunstig", "Vrijwilligers vervullen daarin een onmisbare rol", "De regering ondersteunt deze innovaties", "Natuur en landschap maken de leefomgeving aantrekkelijk"
2001	"Onze samenleving verandert snel", "Internationale samenwerking staat onder druk", "Omvangrijke investeringen in bereikbaarheid blijven noodzakelijk", "Leden van de Staten Generaal", "Zorg op maat staat voorop"
2002	"Die waarde moet behouden blijven", "Leden van de Staten Generaal", "De internationale conjunctuur is ingezakt", "Er komt een landelijke recherche", "Nederland kent een grote traditie van internationale solidariteit"
2003	"Leden van de Staten Generaal", "Veelplegers worden met voorrang aangepakt", "Dagelijks worden honderden mensen werkloos", "Hiertoe zijn heldere rechtsnormen geboden", "Het aanbod van inburgeringcursussen wordt vrijgelaten"
2004	"Leden van de Staten Generaal", "De wereldeconomie groeit in hoog tempo", "het integratiebeleid moet hieraan bijdragen", "Vertrouwen geeft een samenleving veerkracht en daadkracht", "Op rust een verantwoordelijke en zware taak"
2005	"Solide oplossingen vragen tijd", "De criminaliteit daalt", "een veiliger Nederland", "Deze ingrepen doen zeker pijn", "Jonge dynamische economien komen op"
2006	"Nederland werkt", "Cultuur verbindt en verrijkt", "Mensen voelen zich veiliger", "Kinderen worden kosteloos meeverzekerd", "Nederlandse ondernemers zijn wereldwijd actief"
2007	"Jongeren vinden snel een baan", "Nationale parlementen krijgen een sterkere rol", "Die gevoelens kunnen diep ingrijpen", "Ondernemers en consumenten zijn optimistisch", "De staatkundige verhoudingen worden herzien"
2008	"de motorrijtuigenbelasting verhoogd", "Wie kan moet meedoen", "Alertheid blijft echter geboden", "Vrede en veiligheid vragen voortdurend aandacht", "De werkloosheid is laag"
2009	"Leden van de Staten Generaal", "Dit mogen wij niet laten gebeuren", "Hiermee wil de regering de economie stimuleren", "Deze waarden vinden in Europa hun oorsprong", "Daardoor zullen meer werknemers hun baan behouden"

2010	”Schooluitval moet effectief worden bestreden”, ”Leden van de Staten Generaal”, ”Hierbij staan kwaliteit en toegankelijkheid centraal”, ”De overheid schept hierbij randvoorwaarden voor duurzame productiemethoden”, ”Onze stijgende levensverwachting is een groot goed”