The EU Parliament in clouds

An applied study to the usability of word clouds to summarize meetings and the methods to construct them

Gilles de Hollander
0586544

Bachelor thesis
Credits: 9 EC

Bachelor Opleiding Kunstmatige Intelligentie

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisor
dr. M. Marx

Information and Language Processing Systems
Informatics Institute
Universiteit van Amsterdam
Science Park 107, 1098 XG Amsterdam,
The Netherlands.

June 25th, 2010

Abstract
In this study parsimonious language models were used to construct word clouds of the proceedings of the European Parliament. Multiple design choices had to be made and are discussed. Important features are stemming during tokenization, including bigrams into the word cloud and multi-lingualism. Also, the original parsimonious language models were extended with an additional term dampening unigrams that already occurred in the word cloud.
This algorithm was tested in a small user study, using proceedings of the Science faculty's student council. Members of this council had to give their preference for multiple word clouds constructed using either parsimonious language models or simple TF with stop words. 68% over 29% ($p < 0.05$, two-tailed paired t-test) preferred the word clouds constructed using parsimonious language models.
Beside the system design further technical findings, the social significance of applying word clouds to political data and possibilities for future work are discussed.

Table of Contens

1. Introduction

1.1 A Navigation Tool for the European Parliament

This paper will describe and evaluate a proof-of-concept system that makes navigating through the proceedings of the European Parliament (or possibly other political entities) easier, faster and more joyful by applying a technique called *word clouds*. Word clouds are a representation of a document that gives a very quick, visual impression of the document's subjects. A system like this could help make European citizens more concerned and involved with European politics.

The idea to apply word clouds to proceedings of the European parliament is largely inspired by the website capitolwords.org, where something very similar is done using the proceedings of the United States Congress: it counts the number of times words occur in speeches in congress. In this way the Sunlight Foundation[1], which is after this website, tries to 'open up' democracy, by making it easier for citizens to see which subjects are discussed in Congress, what the trends are in these subjects and with which subjects specific politicians are occupied with.

It has been tried before to build a website around word counts of proceedings of the European Parliament by some pre-master information science UvA students (Besseling, Oudshoorn, Theis, & Visser, 2010). Their project was called 'wEUrds.nl'. In this project it turned out to be hard to treat these large amounts of data fast enough. Their project was, alas, also not very well documented, making it hard to learn from it.

The main extension of my EU-system to capitolword.org and the wEUrds.nl-project is that more advanced models are used to construct the word clouds. Not simple counts with a 'stop word'-list containing words that are not meaningful enough, but, drawing upon the work of (Kaptein, Hiemstra, & Kamps, 2010), more advanced *parsimonious language models* are used, explicitly modeling the 'discriminative value' of terms for a document from its corpus. The original extended with a small feature making including bigrams into the word clouds easier.

[1] http://sunlightfoundation.com/

Another interesting additional feature to the original capitolwords.org-project is that, as EU parliament proceedings data made this possible, the word clouds of the different proceedings can be shown in multiple languages.

And, last but not least, using the right XML-schema, proceedings from any political entity could be processed into word clouds. With this system that has been shown to work effectively with even a small corpus of proceedings of meetings of the faculty's student council.

To describe and evaluate the system, the following questions will be discussed:

1. **What are word clouds and what are possible applications?**

In section 2 I will discuss what word clouds are, which applications and functions of them can be found in literature and why it is interesting to apply them to proceedings of the European parliament. This will discussed both from an applied scientific view, as these proceedings have some unique properties that distinguish them from many other documents, as the social significance of doing so will be discussed.

2. **How can word clouds be automatically constructed from documents and a corpus and which methods are most effective in doing so?**

In section 3 I will discuss using which methods word clouds have been constructed before and why. In section 4 I will then extensively explain how my system works, which methods it uses and why these methods were chosen and qualitatively evaluate these choices. I will also describe and evaluate two extensions made to the original algorithm by (Kaptein, Hiemstra, & Kamps, How Different are Language Models and Word Clouds?, 2010) and qualitatively compare the system as a whole to the capitolwors.org system.

3. **How do users currently actually rate the usability of word clouds?**

In section 5 the results of a small user study will be discussed:

This study will show the preference of users for word clouds constructed using either 'simple' Term Frequencies with a stop word-list or parsimonious language models and test the hypothesis that these last ones would be more effective/preferred.

2. Word Clouds

2.1 What word clouds are and what they can do

Word clouds are a compact visual representation of a document, where the semantically most distinctive words of the document are shown together in a *cloud* of words. The more distinctive a concept is for the document, the bigger it is in the cloud. The cloud offers a user a method to very quickly get an impression of what the document is about.

Word clouds are quite similar to the well-known *tag clouds* that can be found on many large *web 2.0*-websites. Tag clouds are also built from collections of words with different sizes that should give a representation of a document. The crucial difference is however that tag clouds are built by users: they 'tag' documents by giving them certain tags, representing their semantic

meaning. The tag clouds are then constructed using not word counts but number of allocations to the words of the specific tags.

Using word clouds does however offer some advantages over using tag clouds. Firstly word clouds of course do not need any human interpretation and tagging, making it much easier to make clouds for large corpora. Also, tag clouds are built by using tag associations of specific (parts of) documents, making it possible to only make clouds for exactly those parts or sets of these parts. Word clouds can be made from almost any subset of a document, as long as this set is still large enough to form a meaningful word cloud. And, last but not least, word clouds refer to actual words that must actually occur in the document(s), making the words in the cloud an ideal entry-point to a search of a specific word in the document(-set) and making it possible to highlight it.

User-generated tag clouds are however much more studied and present on the web than word clouds, their automatically-generated brothers, probably because they are easier to make. The findings about tag clouds are still very relevant though. According to (Hearst & Rosner, 2008), tag clouds are very popular, mostly because they are 'fun and non-conformist' and a 'social signaler' and not so much because they would offer that good help in information processing tasks. Something (Rivadeneira, Gruen, Muller, & Millen, 2007) have empirically shown: an ordered list often works better to help someone quickly find what he/she is looking for.

According to (Rivadeneira, Gruen, Muller, & Millen, 2007) tag clouds can have however more functions besides locating a specific term that represents a desired concept, they can also be used in:

- Browsing: casually explore documents using clouds with no specific target in mind.
- Impression formation or gisting: use the clouds to get a general idea on the underlying data.
- Recognition / matching: recognize which of several sets of information the tag cloud is likely to represent.

These functions are interesting to keep in mind when applying word clouds to proceedings of for example the European Parliament: it makes possible to think about the non-conformist, informal way of browsing through these proceedings the system should offer. With 'social signaler' Rivadeneira et al. also offer an interesting concept, as political speeches are probably also quite susceptible to trends and hypes that, using word clouds, can be observed in an easy and fun way.

2.2 Word Clouds to navigate through European politics

In this study we applied more advanced word cloud-techniques to form an impression of the proceedings of the EU parliament. We primarily did so as the semantic contents of this data is especially interesting to analyze using word clouds and we wanted to lay the foundation for a system that could make European citizens more concerned and involved with European politics.

However, also from an applied scientific view on how word clouds can be improved, these documents are, compared to many others, especially interesting, for multiple reasons:

1. Firstly they have a clear dimension of time: they follow each other in sequence and it can be interesting to see how word clouds 'move trough time' as different subjects appear and are handled in series of debates;

2. The documents are annotated: from every word in a proceeding it is known which member of the parliament has said it. This makes it possible to also make word clouds that are clustered not by proceeding but by Member of Parliament. And, because the EU has an easy accessible database linking members of parliament to their fractions, their parties and their countries, clustering by these three facets is straightforward as well. Giving the word cloud the possibility of performing a more social function as described in the preceding section;

3. Political documents are relatively hard to scan for the main concepts manually, as political documents tend to be long and contain a lot of jargon. For such documents, the 'Impression Formation'-function of word clouds comes in extra useful;

4. There is a lot of data, making the corpus to which concept-frequencies can be compared larger, offering the possibility of very fine-tuned word values.

5. The documents are all multilingual, making comparisons of word clouds in different languages possible

Social Significance

As suggested, the system described in this study could also socially be very significant. It can offer a way to make it easier for European citizens, especially the younger technology-minded, to become more involved with European politics. It fits well among other projects of the 'Political Mashup'-research program of the UvA (Political Mashup, 2010) that tries to use technology to make politics more easily-accessible, also the less-known parts, like European politics (Nusselder, Peetz, Schuth, & Marx, 2008).

Multiple studies have indicated that especially younger people are more interested in politics than often assumed by the general public ((Aalberts C. , 2006), (Aalberts C. , 2004), (Costera Meijer, 2006), (Gebuis & van Hoof, 2010)). They are especially concerned with the content of the political debate, but do wish to be informed by other means than the traditional information sources. They wish to be informed by 'post-modern information sources'.

This concept of 'post-modern information sources' is developed and tested in (Costera Meijer, 2006) and (Gebuis & van Hoof, 2010): their theory states that these 'post-modern information sources' differ from traditional ones on the following:

1. Experience instead of knowledge and insight
2. To participate instead of beholding

3. Images instead of text
4. A feel of connection with others instead of individualism
5. Game, chance and anarchy instead of goal, design and hierarchy

It can be argued that a system like the one described in this study could very well satisfy these directives: The playful nature of the experience of 'exploring' something as serious as the proceedings of the EU Parliament really satisfies directives number 1 and 5. The possibility to see what individual members and fractions of parliament are occupied with what subjects in this way satisfies directive 2 and 4 quite well and, of course, the word clouds are indeed a very 'image-like' way of presenting the information, fitting directive 3.

In a broader sense, the system built in this study could also be applied on other large data sources. For example, to keep in the domain of politics, older proceedings of the national parliament, like the one indexed by (Marx & Gielissen, 2009): these collections are extremely large, making manual scanning of texts very time-consuming and good 'meta-data' like word-clouds extremely useful.

3. What word clouds can do, how they do it and what can be improved

As has been said, word clouds are not yet very well-studied in literature or present at the web. Word clouds, unlike tag clouds, have to be automatically generated, which turns out to be quite hard. Some approaches have been used, relying heavily on what is known from the information retrieval literature, which is mostly concerned with finding documents giving a query. For this they use models like $P(D|t)$: given a term t, how relevant is document D in corpus C. With these models a search algorithm can now look for the Document with the highest relevance given a set of terms (query).

You could approach creating word clouds as the inverse of search, where instead of documents given a term, it's the terms that have to be found, given a document within a corpus. So now we have a document D and ask which terms t are most relevant: $P(t|D)$.

In this section I will discuss some approaches to creating word clouds and how they could be improved.

Simple TF and its problems

The simplest approach to create word clouds is, of course, to just count the words in a document and let their frequency determine if they are included in the word cloud and what will be their size. Essential is to ignore so-called 'stop words', often functional words like 'the' and 'or', otherwise almost the entire word cloud will be filled with them. This 'TF' (term-frequency)-approach is quite conventional and widely used. The popular websites wordle.net (Feinberg, 2010) and 'Many Eyes' (IBM, 2010) use it, as well as the inspiration for this project, capitolwords.org. This TF-approach works quite well, but leaves a lot of room for improvement:

Firstly, words that are semantically similar but have different syntax are considered as different concepts: in this way a word cloud can contain both 'camel' and 'camels' or 'disappear' and 'disappearing', wasting room for more informative words.

This problem can be solved using so-called ‘stemming’-algorithms, first developed for English already in 1979 by (Porter M. , 1980), that stem words to their most basic form. A document then can be completely stemmed, these stems can be counted, after which they can again be de-stemmed, showing a meaningful form of the stem as a word in the word cloud.

This stemming and de-stemming however often also turns out to be problematic. It can be too ‘aggressive’ and stem away important semantic information: ‘ouderen’ to ‘oud’ (‘elderly’ to ‘old’ in English) is the usual example in Dutch. Our system tries to overcome this by looking which de-stemmed variant of a given stem occurs most in the given document.

Figure 1 Very soft stemming: tutor gets removed and just tutoren stays, but twee and tweede stay

A second problem is that concepts are often not represented by just one word. A simple example would be names: ‘John Smith’ is more informative and unique than either ‘John’ or ‘Smith’. A simple TF algorithm cannot establish that John and Smith are always together in a document. A solution to this would be to also count all occurring bigrams and treat them just like unigram. In this way very discriminative bigrams can also occur in a word cloud. (Kaptein, Hiemstra, & Kamps, How Different are Language Models and Word Clouds?, 2010) have found that users do prefer these word clouds containing bigrams over word clouds not containing them.

Figure 2 Word Cloud with and without including bigrams like 'Research programme' and 'Framework Research

A third and central problem with this approach is that it is hard to determine whether a word is a stop word or not, especially within a given domain (e.g. 'subsidy' or 'commissioner' for European politics): this border is quite an arbitrary one. Of course functional words (on their own) are almost never too informative, but so are words like 'chairman', 'proceeding' or 'commissioner' for proceedings of the EU Parliament. Where to put this border? It is often very labour-intensive and complicated to create domain-specific stop word-lists, so one might rather automate this process.

TF-IDF

To automate the stop word-problem, more advanced methods than simple TF should be used: it should not be up to the user to manually indicate whether words should or shouldn't be included in a word cloud *a priori*, but these should be selected automatically, based on the corpus (domain) the document is in. A bit more complicated sister of the TF-approach that can do this, is the TF-IDF, or *Term Frequency Inverse Document Frequency*-approach. Here, an important part of the discriminative value is the IDF: the log of the inverse of the relative number of documents that contain the term:

$$IDF_t = log \frac{|D|}{1 + |\{D \in term\ t\ occurs\ in\ D\}|}$$

For TF-IDF the term frequency is now multiplied with this IDF, leading to very small values when a term occurs in (nearly) every document and larger values when a term occurs in only a fraction of the documents.

Language models

(Kaptein, Hiemstra, & Kamps, How Different are Language Models and Word Clouds?, 2010) found that although TF-IDF is an elegant technique that can work well for information retrieval, users still prefer simple TF-word clouds (with stop words-list) over TF-IDF word clouds. In the same study a more complex form of language models, *parsimonious language models*, developed by (Hiemstra, Robertson, & Zaragoza, 2004) was used to create discriminative values with models that explicitly model the way the occurrence of words in a document differs from its corpus. This model showed more promising results and offered both parameters and extensibility that made it an interesting algorithm to use. It was this model that was slightly modified and used in the system, incorporating the stemming- and bigrams-solutions described above. A technical description of these solutions and the parsimonious language models can be found in section 4.4.2.

4. System Design

4.1 The Goal

The main goal of the system was to support a website that makes it easy and enjoyable for European citizens to quickly see what subjects the European Parliament is talking about, using its proceedings. The system had to use modern, playful 'web 2.0' techniques that would appeal to a more technologically-minded public. Therefore the technique of word clouds (see section 1.2) was chosen to present this information. These word clouds are based on the actual words and their frequencies in the proceedings of the European parliament.

Another underlying goal was therefore to make it easier for European citizens to browse European proceedings. Using the word clouds as meta-data that can give a quick but representative impression of a given (set of) proceeding(s) ('impression formation' in the words of (Rivadeneira, Gruen, Muller, & Millen, 2007)), European citizens had to be able to use the system to find passages in proceedings that are meaningful to them in a fast, intuitive way.

Also, the system was to make it easier for citizens to see which members of parliament are occupied with which subjects and what they are saying about it. In this way citizens can easily see which members of parliament are occupied with subjects that are meaningful to them, and, the other way around, which subjects a member of parliament they voted for or they are otherwise interested in is occupied with. This can maybe close a bit of the gap between citizens and members of European Parliament that is often referred to (for example in (European Parliament, deparment The Netherlands, 2010).

Lastly, the system had to be built in such a way that it could easily be used with proceedings of other entities that use the same DTD (a scheme for how XML-documents representing proceedings should be formed).

With these goals in mind, a functional design for the system was made and will now be described in the rest of this section.

4.2 Functionality

Word clouds

The most basic functionality of the system is making word clouds based on speeches in the European Parliament, or other. These word clouds are constructed by comparing the relative frequencies of words and bigrams in a given set of speeches, as compared to the entire corpus of EU speeches in the dataset. There a multiple ways this set of speeches that will be represented in a word cloud can be brought together: the set here representing a certain common property of the speeches.

Firstly, the speeches can be clustered by proceeding: in this way it's possible to see which words were most distinctive for a given debate (day) or even a longer period of time, as the clustered information can easily be obtained from multiple proceedings within a given time frame. The system thus makes it possible to make word clouds for specific days, weeks, months and years. The website makes it easy to navigate through these different periods.

Secondly, the speeches can be clustered by Member of Parliament: this gives users the possibility to see which words are most distinctive for a given member. Using the same information the same can achieved the other way around: a user can put in a word and see in the speeches of which member they relatively occur the most.

Bar Graphs

The information that is present in the word clouds can also be presented in a somewhat different form: a bar graph-view, with graphs indicating the relevance of the term in it.

Search

By clicking on a word in a word cloud or barchart, the proceedings can be searched for this term: the speeches in which it occurs (the most) will be shown, as well as where in the speech they occur. It is also possible to search in the more traditional way of entering terms or adding additional terms in the word cloud.

Trend Graphs

The system can make timelines: graphs of a term, indicating how distinctive it was for individual proceedings in a given sequence proceedings (timeframe). This can also be done for multiple terms at the same time, giving a comprehensive overview of when certain related subjects were talked about.

Members of Parliament

Every member of parliament has a little page on the website with some biographical information, his/her personal word cloud and a link to his personal webpage. In this way it's easy for the user to find more information about members of parliament that are occupied with a given subject.

Countries

As it is known for every member of parliament which country he/she represents and which words were spoken by them, it's possible to make separate word clouds for given periods of time per country. In this way it can be seen which subjects are important in which countries in a given period of time. Also, given a subject, it can be seen member of parliament of which nationality are most occupied with it. This information will all be visualized using a so-called 'heat-map' of Europe.

Proceedings

As the main data that is used by the system is of course the proceedings of the European parliament and as these can be used, driven by the first impression of a word cloud, to get a more detailed account of what is said in the European Parliament, an important function of the system is serving these proceedings. Users can browse through them using the system, exploiting the structure of topics and speeches within the proceedings to make this as easy as possible.

Multilingualism

The European Parliament is of course a multilingual institution. The proceedings are therefore always made in a wide array of languages. The techniques that are used in this system are easily extended to another language. The system makes it possible to view the proceedings and the word clouds based on them in multiple languages: for now just English and Dutch but with just a few very small adjustments any of the official EU Parliament languages can be used.

Figure 3 Multi-lingualism: two word clouds of the same EU Parliament proceeding in both Dutch and English

4.3 The Data

XML

The system uses, as original data, the proceedings of meetings of the EU Parliament back to the 20th of July 1999 until the 20th of May 2010. The proceedings are translated in at most 21 languages. The newest proceedings are not yet translated in 'smaller' languages and the older proceedings are not yet translated into languages of new member states.

The proceedings are annotated with some relevant meta-information about that which is spoken, in XML-format and accessible with an open-source XQuery database server, eXist-db (eXist-db, 2010). The XML-formatted annotations make it possible to know for every word that is spoken during an EU Parliament debate:

1. The 'document number', which is the index that the EU Parliament uses to index their proceedings
2. The date of the debate the word was spoken on
3. During which subtopic of the debate the word was spoken
4. Who has spoken the word

For European politicians there also is a free database, offering the opportunity to also know for every speaker of a word in the EU Parliament:

1. The national party and the European parliemant fraction of the speaker
2. The EU member state the speaker is representing
3. The role in which the speaker is speaking (e.g., chairman of parliament, member of the European Committee)

A search-engine for colelctions in the schema used for the EU parliament is available with the following options

This database also offer the opportunity to search the EU Parliament proceedings for speeches:

- Containing a specific string
- from a specific Member of Parliament (by ID or name)
- from a specific period

Python

This original data, that keeps expanding after every debate of the EU Parliament, can be analyzed daily by a python script during off-peak hours, to create the data that the website uses to construct the word clouds. For this, every relevant proceeding is loaded and processed with Python from the XQuery-server and converted into simple 'Proceeding'-Python-objects that contain its texts, annotations, term frequencies and language-model values, constructed by comparing the term frequencies to the other proceedings in the dataset.

The proceedings can then in turn be used to create 'Member'-Python-objects that represent members of parliament and contain every speech a member has spoken. These objects also contain term frequencies and possibly other language-modeled values that indicate which terms are most distinctive for a member compared to the other members.

MySQL

To make it possible to present the information created by Python, it is inserted into a MySQL-database that is easily and quickly accessible with php, that is used to render the website. Python stores its analysis in the following tables:

- **proceedings:** Containing important information about proceedings: mainly their date and documentation number.
- **topics:** Containing topic titles and to which proceeding they belong.
- **speakers:** Containing information about speakers
- **wordcloud_proceedings:** containing the pre-calculated values that can be used to build a word cloud per proceeding/a group of proceedings.
- **wordcloud_topics:** containing the pre-calculated values that can be used to build a word cloud per topic/a group of topics.
- **wordcloud_speakers:** containing the pre-calculated values that can be used to build a word cloud per speaker/a group of speakers.

4.4 The Algorithm

To construct the word values for the word clouds, given proceedings-data, the following steps are performed for every proceeding:

4.4.1 The counting of terms

First, the terms occurring in every proceeding are tokenized and counted. The term are can both be individual words or bigrams. These terms are stemmed by the Snowball stemming algorithm (Porter M. , 2010) corresponding to the language of the document they're in. In this ways terms that are only syntactically different but mean the same thing (e.g. 'subsidy' and 'subsidies') are grouped together. If stems occur more than once in the proceeding, they are also counted for the entire corpus.

For every stem and bistem the number of times corresponding words and bigrams occurred are also kept. In this way it's possible to see how many times terms with a given stem/bistem occur and which term with this stem/bistem is most frequent. The word that is most frequent in a proceeding will later be used in the word cloud to represent that specific proceeding.

(Kaptein, Hiemstra, & Kamps, How Different are Language Models and Word Clouds?, 2010) suggested this as an improvement over their own system where the stem/term-counts were counted only for the corpus as a whole. As it sometimes can occur that terms with the same stem mean different things and occur with these different meanings in different proceedings, the method used here can probably give semantically more representative word clouds.

4.4.2 The comparison to the corpus

As a second step in the process, the frequency of terms in every proceeding and its topics and speeches is compared to their frequency in the entire corpus and a value is calculated that determines which terms will end up in the corresponding word clouds and which size they will

have. For this a parsimonious term weighting scheme is used, developed in (Hiemstra, Robertson, & Zaragoza, 2004) and applied to word clouds before in (Kaptein, Hiemstra, & Kamps, How Different are Language Models and Word Clouds?, 2010) an (Kaptein & Marx, 2010). As I slightly extended their approach for this system, I will now describe their approach and which modifications were made.

Parsimonious language models: existing approach

Central to parsimonious language models is the idea of parsimony: in information retrieval one is basically only interested in terms in a document that distinguish it semantically from other documents in the corpus. In the usual bag-of-words approach, terms that occur often and are distributed evenly over the corpus offer virtually nothing of this. Parsimonious models exploit this fact and 'greedily' throw these 'common' non-discriminating terms away, leaving more room in the probability mass for 'more interesting' terms. This offers computational efficiency, but also more precision in computing which terms are most discriminating.

These parsimonious language models do so using the following EM-algorithm:

1. **E-step:**

$$e_t = tf(t, D) \cdot \frac{\lambda P(t|D)}{(1-\lambda)P(t|C) + \lambda P(t|D)}$$

Equation 1 E-step old parsimonious model

2. **M-step:**

$$P(t|D) = \frac{e_t}{\sum_t e_t}$$

Equation 2 M-step old parsimonious model

This algorithm is repeated a fixed number of steps (100) until (near) convergence. After every M-step terms with a $P(t|D)$ below 0.001 are considered non-discriminating and removed. $P(t|C)$estimates the relative frequency of the term in the entire corpus of proceedings (with a maximum likelihood estimate).

The goal of this algorithm to get a good estimation of $P(t|D)$, $P(t|D)$ being a model of how likely a document D can 'produce' a query containing a term t, or: when someone is looking for document D, how likely is it that he/she will use t in a query?

The algorithm removes non-discriminating terms from the $P(t|D)$model that are already 'explained' by the corpus model $P(t|C)$. The λ-parameter determines how much of the occurrence of a word may be explained by the document model. The lower λ gets, the more unique the words that remain will be. This parameter can of course be too low: then only very unique words remain that are probably too special to indicate what a document is about.

The rationale behind this is that if someone is looking for a specific document or wants to describe it using a specific term, this term will not be the term that is used most in a document ('the' or 'commissioner'), but a term that is 'discriminative' but still 'common': it occurs in the entire corpus, but mostly in the specific documents about it ('Bulgary').

Extended approach

Bigram Model

In my extended approach words and bigrams were modeled separately: both are stemmed, but counted apart. This was done because an extra λ_2-parameter was introduced, to try to exclude words that are already included by the bigrams in the word cloud. In the original model it can easily occur that both "John", "Smith" and "John Smith" end up in a word cloud. To just show "John Smith" is semantically more elegant and leaves room for 2 more terms actually presenting new, discriminating information. To try to achieve this, the λ_2-parameter was introduced. This parameter determines how much of the occurrence of a term should be explained by it already occurring in a bigram. In this way, the $P(t|D)$ of a term will be much lower if it (almost) exclusively occurs in a bigram with a high $P(t|D)$.

The formula used for unigrams was now thus:

1. **E-step:**

$$e_t = tf(unigram, D) \cdot \frac{\lambda_1 P(unigram|D)}{(1-\lambda_1-\lambda_2)P(unigram|C) + \lambda_1 P(unigram|D) + \lambda_2 P_{bigram}(unigram|D)}$$

Equation 3 E-step new parsimonious model

2. **M-step:**

$$P(unigram|D) = \frac{e_t}{\sum_t e_t}$$

Equation 4 M-step old parsimonious model

And where:

$$P_{bigram}(unigram|D) = \sum_b P(b|D)$$
$$where\ unigram\ \epsilon\ bigram\ b$$

Equation 5 M-step old parsimonious model

…where $P_{bigram}(unigram|D)$ represents the parsimonious probability that the unigram $unigram$ occurs in a bigram in the word cloud. It is the sum of the parsimonious probabilities of the bigrams calculated with the classic parsimonious model (equation 1 and 2) that contain the specific unigram.

An important note to make is that when λ_2 is set to 0, the new formula does the same as the old formula and the $P(t|D)$)-values of the bigrams have no effect on the $P(t|D)$-values of corresponding words. Also, whether the words and bigrams were taken together or apart to build their language models had virtually no effect on their final $P(t|D)$-values. This is a good thing, as apparently it offers the possibility of using the λ_2-parameter without doing harm to the success of the original model.

If λ_2 was increased, unigrams that already occurred in a relevant bigram occurring in the word cloud got a high$\lambda_2 P_{bigram}(unigram|D)$ and thereby a low $P(t|D)$ and would disappear from the word cloud. However, the effect of the λ_2-parameter on the occurrence of these superfluous unigrams differed across corpora of different size and source and was influenced by the choice of λ_1 and the bigram bonus, making it hard to find the right λ_2-parameter setting. More work on the exact nature of this could be fruitful.

Bigram Bonus

(Kaptein, Hiemstra, & Kamps, How Different are Language Models and Word Clouds?, 2010) showed that bigrams can be more meaningful than unigrams and semantically discriminate a document from its corpus. To further test this hypothesis for my system, it has the possibility to give bigram terms a *bigram bonus*: a factor the (parsimonious) values indicating the discriminating quality of bigrams is multiplied with.

4.4.3 From model to clouds

When this system would be scaled up, it would computationally be very inefficient to recalculate the parsimonious language models every time the web server receives a page request. That is why their results are stored in a MySQL-database which the web server uses to construct the word clouds.

This means that for every proceeding, topic and speaker, a set of 50 words and their distinctive value, according to the parsimonious models, is inserted in corresponding tables. These 50 words are more than needed for the standard clouds of size 30, but this leave some room for increasing their size and clustering over multiple clouds. By clustering multiple parsimonious models of multiple documents it possible for the frontend web application to create word clouds for specific proceedings, topics and speakers, but also groups of them. By summing these values up over multiple proceedings or speakers and combining these with meta-data about them, word clouds can be constructed that represent specific periods of time or groups of speakers (nationality/fraction).

4.4.4 Choosing the Parameters

It turned out to be hard, a matter of taste and depending on circumstances what the 'right' parameters, leading to the most representative word clouds are. Also, as the system was also applied on proceedings of the faculty's student council (FSR FNWI), the influence of the nature of the corpus could be tested. It turned out that this and the length and properties of the document that is compared to it have an (undesirably) important effect on which parameters seem most fit.

Also, as a corpus always has a lot more unique bigrams (in the order of quadratic) than words, making incorporating bigrams into word clouds a bit complicated. Here I will discuss some finding in choosing the right parameters.

- $\boldsymbol{\lambda_1}$: for the FSR proceedings, with around 5490 unique stems a value of around 0.001 – 0.01 seemed very fitting and was qualitatively evaluated positively by multiple users. For

the EU proceedings, based on a much larger corpus, these values made the clouds contain only single parliament members and very specific words. There λ_1 up to 0.5 seemed effective.

It also seemed that topic-clouds become more effective when λ_1 has higher values.

- **λ_2**: It turned out to be very hard to find right values for λ_2, this could both be interpreted as a large disadvantage of the 'unigram-dampening' or an incentive to find the dependencies that rule how this mechanism works.
- **Min Number of stems:** It turned out that documents (e.g. speeches) can be too short to create a meaningful word cloud: the number of unique words a cloud is based on should be at least the double of the number of words that end up in the cloud. In my system I chose 50 as a good estimate.
- **Bigram bonus proceedings:** Could especially matter of taste. It turns out that quite quickly when the bigram bonus is set above 1, bigrams like "X says" are selected, which might be not that informative. A value of 2 however does often leads to the inclusion of informative-and-otherwise-excluded bigrams.
- **Truncation:** in the EM-process, terms with a low parsimonious value below this threshold are removed. (Kaptein, Hiemstra, & Kamps, How Different are Language Models and Word Clouds?, 2010) used 0.001. However, in our system, it turned out that larger proceedings could contain so many unique bigrams that in the first step every one of them would be thrown away. So to include bigrams in the final word clouds, a value of 0.0001 seemed more fit.

4.5 The Frontend

The interface of the system is a small website that is easy to navigate trough. When one enters this website one can see the latest weekly word cloud and most vocal parliament members; clicking them leads to their personal page on the website. There is also a large bar that makes it possible to navigate through the main features of the system:

- Browse through the word cloud and statistics per proceeding
- Browse through the different speakers of the parliament and their word clouds and other statistics
- Make a word cloud for a specific period of time
- Search trough the proceedings for specific terms
- Change the language

4.6 Differences in functionality to capitolwords.org

The system was heavily inspired by the American website capitolwords.org and has tried to both honor this inspiration and extend its ideas with some improvements which will be discussed here.

First and foremost does capitolwords.org use 'simple' term frequencies with a stop-word list to construct its word clouds. Although these frequencies can give a clear indication what is talked about most, they can be terms that are very common in a lot of debates

('commissioner', 'member of parliament') and not so much the terms that are central to the debate ('immigration', 'economic crisis'). Of course stop word lists can be extended to fit the proceedings of a specific institution, but that does not really solve the problem too elegant or general. My EU-system uses language modeling techniques to try to prevent this. In the user study (next section) we have tried to find out if users would indeed value this extension from a simple term frequencies to a language modeling-approach.

A second difference is that the European proceedings are multilingual and so is the system. It's possible to browse word clouds and proceedings in different languages. Maybe even adding another bit to the playful aspect of the website and purveying something for people speaking multiple languages to look at.

Thirdly, the coupling to the proceedings is a bit tighter in the EU-system: the proceedings are browsable on the same website and they can be searched, also by clicking terms in the word clouds.

Of course there are countless other differences superficial differences that will not be discussed here, especially as capitolwords.org is a real-life foundation and not the result of a bachelor project.

5. Empirical evaluation of parsimonious language models

5.1 Research question

A user study was performed to assess the system representing proceedings of meetings using parsimonious models. The main research questions of this study was the following:

How do users value word clouds of proceedings of meetings they attended constructed with the extended parsimonious language models, as compared to those constructed using simple term frequencies (TFs) with a stopword-list?

5.2 Method

To see if word clouds constructed using either parsimonious language models in my system or clouds constructed with TF with a stop word-list are most effective, a group of *12* students, all members of the faculty student council, have indicated their preference for clouds constructed with meetings of the student council they attended. They rated 7 pairs of clouds representing entire meetings and 7 sets of pairs of clouds representing the topics of a given meeting. They had an extended instruction to choose the word cloud that gave the most best representation of what the meeting was about and could give someone who did not attend the meeting a good impression (see Appendix A for the exact text in Dutch).

The pairs always consisted of one *TF* and one *parsimonious* word cloud, their sequence being randomized every trial. The students were asked to choose the cloud that they believed gave them the best indication which subjects were most distinctive for the given meeting or topic and would give someone that had not attended the meeting the most representative impression of what had been said.

The parsimonious word clouds were constructed using a λ_1 of 0.01, a λ_2 of 0.001, no bigrambonuses and a minimum number of unique 50 stems per topic. The clouds contained 25 terms.

5.3 Results

Word clouds constructed using the parsimonious model were significantly more preferred than those constructed using TF with a stop word-list. P was <0.05 for a two-tailed paired t-test (n=12).

Document(-part)	Pref. pars. model (SD)	Pref. TF model (SD)	No preference (SD)
Topics	65 % (25%)*	32% (26%)*	3% (6%)
Proceeding	71 % (24%)*	25% (25%)*	4% (9%)
Total	68% (17%)*	29% (20%)*	3% (6%)

Table 1: results of user study, percentage of preference and SD

5.4 Conclusions

Parsimonious language models turned out to largely be preferred over their more simply constructed TF-sisters. Also, qualitatively, participants of the study often indicated they preferred word clouds excluding semi-functional words like 'voting' or 'thinking', indeed words parsimonious models filter out more easily. Also, they often enjoyed browsing through the word clouds and suddenly remembering certain parts of meetings they attended.

6 Discussion and future work

In this study we applied and extended the idea of (Kaptein, Hiemstra, & Kamps, How Different are Language Models and Word Clouds?, 2010) to use parsimonious language models to construct word clouds on political proceedings, inspired by the simple word count-site capitolwords.org. We thereby showed that the work of Kaptein et al. can be applied in a more practical real-world setting.

We have also discussed how word clouds can be used in more settings like these, offering the possibility for users to quickly browse through large sets of documents and get an impression of their contents. This 'post-modern', visual way of information representation could even be used as a marketing tool to increase people's interests in a specific domain in a substantive way, something a more streamlined future EU-system could do for European politics.

We showed how our system is designed and which choices were made. Especially estimating the parameters of the models turned out to be hard and depending on the corpus and even taste and individual documents that are compared to it.

All in all we did however build a working system which showed promising word clouds which were shown in a small user study. Also these users saw potential in the application of word clouds on political proceedings and quantitatively, blindly assessed that word clouds

constructed using parsimonious language models are indeed even more representative than clouds constructed using TF.

Future work

The results of this study could lay on the foundation of a lot of future work. Firstly, the EU system is not ready to actually be used in a production environment. Especially a good interface and clean-up of the behind-the-scenes code could still be a lot of work, but promise a very interesting website.

This could also give the opportunity to perform a larger, qualitative user study of the usability of word clouds: when given an actual website containing these proceedings, how do users rate them and how could the interface, features and the clouds themselves be improved?

On a more fundamental level this system has shown that it is hard to find the 'right' parameters of the parsimonious language model and the bigram-bonuses. Often the values that lead to the most representative clouds differ across different parts of documents (speeches versus entire proceedings) and corpora. Also, especially the λ_2-parameter seems to depend upon the value of other parameters.

To find out more about these dependencies and maybe find some regularities and rules-of-the-thumb, a system could be built that constructs word clouds using different parameters 'on the fly' so that much faster the effects of them can be assessed. First by designers and maybe later in a user study.

Appendix A: instructions to subjects user study (dutch)

Best mede-FSR-lid,
Top dat je me wilt helpen met mijn onderzoek naar Word Clouds: een manier om documenten op een extreem compacte, visuele manier te representeren. In een Word Cloud worden de woorden die het meest kenmerkend zijn voor een document bij elkaar, samen getoond, waarbij ze ook nog groter worden naarmate ze 'kenmerkender' zijn.
Een mogelijke toepassing hiervan zou kunnen zijn dat het mensen een middel biedt om heel snel een indruk te krijgen van waar een document over gaat. Specifieker zou het gebruikt kunnen worden om een indruk te geven van de belangrijkste onderwerpen van een vergadering aan de hand van de notulen.
Een woordenwolk hoeft geen perfecte samenvatting te zijn, maar zou een hele snelle, concrete indruk kunnen bieden over welke onderwerpen een document handelt.
In het testje dat je zometeen gaat doen zul je een veertiental woordenwolken zien van een aantal vergaderingen en agendapunten van vergaderingen waar je waarschijnlijk aanwezig bent geweest. Als je niet aanwezig bent geweest kun je dat aangeven.
De taak die je nu uit moet voeren is telkens aangeven welke van de 2 wolken volgens jou **het best weergeeft waar de vergadering/het agendapunt over ging.** Probeer daarbij vooral in acht te nemen in hoeverre zo'n wolk iemand die niet bij de vergadering aanwezig was een representatievere indruk biedt. Kan deze zo snel zien of het voor zijn doeleinden wel/niet zinnig is om de notulen zelf door te lezen?
Zou je hieronder even je mailadres in kunnen vullen? Dan kunnen we beginnen

Bibliography

Aalberts, C. (2006). *Aantrekkelijke Politiek?* Amsterdam: Spinhuis / Maklu, Het .

Aalberts, C. (2004). Politieke betrokkenheid en politieke sensitiviteit onder jongeren. *Workshop Kwaliteit van het leven en politieke attitudes.* Antwerpen.

Besseling, R., Oudshoorn, K., Theis, S., & Visser, M. (2010). *WEURDS: An Information Retrieval Project.* Amsterdam: UvA.

Costera Meijer, I. (2006). *De toekomst van het nieuws.* Amsterdam: Otto Cramwinkel Uitgever.

European Parliament, deparment The Netherlands. (2010). *Nederlanders en Europa.* Opgehaald van http://www.europa-nu.nl/id/vh93qqnk8atd/nederland_over_europa

eXist-db. (2010). *eXist-db Open Source Native XML Database*. Opgehaald van http://exist.sourceforge.net/

Feinberg, J. (2010). *Wordle.net: Beautiful Word Clouds*. Opgehaald van Wordle.net: Beautiful Word Clouds: http://www.wordle.net

Gebuis, M., & van Hoof, A. (2010). De dogma's van het nieuws voorbij. Het effect van postmoderne nieuwsmedia op waardering, blootstelling en politieke betrokkenheid bij jongeren. *Etmaal van de Communicatiewetenschap 2010.* Gent: vakgroep Communicatiewetenschappen van de Universiteit Gent.

Hearst, M. A., & Rosner, D. (2008). Tag Clouds: Data Analysis or Social Signaller? *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS'08).*

Hiemstra, D., Robertson, S., & Zaragoza, H. (2004). Parsimonious language models for information retrieval. *Proceedings SIGIR* (pp. 178 - 185). New York: ACM Press.

IBM. (2010). *Many Eyes*. Opgehaald van Many Eyes: Tag Cloud: http://manyeyes.alphaworks.ibm.com/manyeyes/page/Tag_Cloud.html

Kaptein, R., & Marx, M. (2010). Focused Retrieval and Result Aggregation with Political Data. *Information Retrieval* , (in press).

Kaptein, R., Hiemstra, D., & Kamps, J. (2010). How Different are Language Models and Word Clouds? *Advances in Information Retrieval: 32nd European Conference on IR Research (ECIR 2010)* (p. to appear). Milton Keynes: Springer.

Marx, M., & Gielissen, T. (2009). Digital Weight Watching: Reconstruction of scanned documents. *Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data* (pp. 25 - 31). Barcelona, Spain: ACM International Conference Proceeding Series.

Nusselder, A., Peetz, H., Schuth, A., & Marx, M. (2008). Helping people to choose for whom to vote. a web information system for the 2009 European elections. *Proceeding of the 18th ACM conference on Information and knowledge management* (pp. 2095-2096). Hong Kong, China: ACM.

Political Mashup. (2010). Opgehaald van http://www.politicalmashup.nl/

Porter, M. (2010). Opgehaald van Snowball Homepage: http://snowball.tartarus.org/index.php

Porter, M. (1980). Program, 14(3). *An algorithm for suffix stripping* , 130−137.

Rivadeneira, A., Gruen, D., Muller, M., & Millen, D. (2007). Getting our head in the clouds: toward evaluation studies of tagclouds. *Proceedings CHI 2007* (pp. 995-998). New York: ACM.