Summarization of Meetings using Word Clouds

Gilles de Hollander

ISLA, Informatics Institute
University of Amsterdam
Amsterdam, the Netherlands
Gilles.de.Hollander@gmail.com

Maarten Marx

ISLA, Informatics Institute
University of Amsterdam
Amsterdam, the Netherlands
maartenmarx@uva.nl

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 multilingualism. 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 University of Amsterdam 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 Term Frequencies (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.**

Keywords: Text summarization, Open Government Data

I. INTRODUCTION

A. 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 [2]. Word clouds are a representation of a document that gives a quick, visual impression of the topics discussed in a document. A system like this could help make European citizens more concerned and involved with European politics. The same techniques are applicable to proceedings of most parliaments in the world as they all have a very similar structure, based on the British Hansard.

The idea to apply word clouds to proceedings of the European parliament is largely inspired by the website capitolwords.org created by the Sunlight Foundation, a non-governmental organization promoting openness of governments. On this site, 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 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.

The main improvement of our system to capitolword.org 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 [2], more advanced parsimonious language models are used, explicitly modeling the 'discriminative value' of terms for a document from its corpus. Their system is extended with a feature which makes including bigrams into word clouds easier. Bigrams are combinations of two tokens, like 'Security Council', and 'post office.

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 XML-schema for meeting notes developed in [1], proceedings from any meeting can be automatically summarized using word clouds. We have shown that our summarization technique works well with even a small corpus of proceedings of meetings of the faculty's student council.

The paper is organized around 3 main questions:

1) What are word clouds and what are possible applications?

In section 2 we will discuss what word clouds are, which applications and functions of them can be found in the literature and why it is interesting to apply them to meeting notes in general and the proceedings of the European parliament in particular. We discuss this both from an applied and a scientific viewpoint.

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 we will discuss methods by word clouds have been constructed before and why. In section 4 we will then extensively explain how our system works, which methods it uses and why these methods were chosen. We qualitatively evaluate these choices and describe and evaluate two extensions made to the original algorithm by Kaptein, Hiemstra, & Kamps [2].

3) How do users rate the usability of word clouds?

In section 5 the results of a user study will be discussed. This study will show the preference of users for word clouds

Maarten Marx acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the FET-Open grant agreement FOX, number FP7-ICT-233599.

This research was supported by the Netherlands Organization for Scientific Research (NWO) under project number 380-52-005 (PoliticalMashup).

978-1-61284-207-3/11/$26.00

constructed using either 'simple' Term Frequencies with a stop word-list or parsimonious language models and test the hypothesis that these last ones yield better summaries and are thus preferred.

II. WORD CLOUDS

A. What word clouds are and what they can do

Word clouds are compact visual representations 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 then 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. Tag clouds are constructed using counts of the occurrence of tags.

Word clouds offer some advantages over tag clouds. Firstly, for the construction of word clouds there is no need for 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 [3], 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 that is empirically shown in [4]: an ordered list often works better to help someone quickly find what he/she is looking for.

According to [4], 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 meetings: it makes possible to think about the non-conformist, informal way of browsing through these proceedings the system should offer. In [4] the authors coin the term 'social signaler' for this, 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.

B. 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 content 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 through time' as different subjects appear and are handled in series of debates;

2. The documents are annotated: for 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 roles in parliament, their fractions, their parties and their countries, clustering by these three facets is straightforward as well. Using such groupings the word clouds may perform the more social function 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.

C. 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 University of Amsterdam [5] that tries to use technology to make politics more easily-accessible, also the less-known parts, like European politics [6].

Multiple studies have indicated that especially younger people are more interested in politics than often assumed by the general public ([7], [8], [9], [10]). 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 [9] and [10]: 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 [11]: these collections are extremely large, making manual scanning of texts very time-consuming and good 'meta-data' like word-clouds extremely useful.

III. 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 [20]. For this they use statistical 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).

One can approach creating word clouds as the inverse of search, where instead of documents given a term, 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 we will discuss some approaches to create word clouds and how they could be improved.

1. 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 [12] and 'Many Eyes' [13] 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 stemming-algorithms, first developed for English already in 1979 by [14], that stem words to their most basic form. As an example, a stemming algorithm reduces the words "fishing", "fished", "fish", and "fisher" to the root word, "fish". A document then can be completely stemmed, these stems can be counted, after which they can again be "back-stemmed", showing a meaningful form of the stem as a word in the word cloud. This stemming and back-stemming however often also turns out to be problematic. Stemmers are based on heuristic rules and thus can make mistakes. It can be too 'aggressive' and stem away important semantic information. Our system tries to overcome this by looking which de-stemmed variant of a given stem occurs most in the given document. Figure 1 shows an example.

Figure 1. Example of very soft stemming: two word clouds of the same document: 'tutor' which is still in the top document gets removed in the bottom document and just the plural 'tutoren' stays. On the other hand, both 'twee' (two) and 'tweede' (second) remain.

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 (combinations of two words) and treat them just like unigrams. In this way very discriminative bigrams can also occur in a word cloud. Kaptein, Hiemstra & Kamps [2] have found that users do prefer these word clouds containing bigrams over word clouds not containing them. Figure 2 shows this by an example.

Figure 2. Two Word Clouds of the same document with and without including bigrams like 'Research Programme' and 'Framework Research'

A third and central problem with the simple term-frequency 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). Of course functional words (on their own) are almost never informative, but so are words like 'chairman', 'proceeding' or 'commissioner' for proceedings of the EU Parliament. It is very labor-intensive and complicated to create domain-specific stop word-lists, so one might rather automate this process.

1) 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 [20]. 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. For C a collection of documents and t a term,

$$\mathrm{IDF}_t = \log \frac{|C|}{1+|\{d \in C \mid t \text{ occurs in } d\}|} \qquad (1)$$

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.

2) Language models

Kaptein, Hiemstra & Kamps [2] 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 [20], parsimonious language models, developed by [15], 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. The technical description of these solutions and the parsimonious language models are described in section IV.

IV. System Design

A. Goals and Functional Requirements

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 [4]), 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 [16].

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. In this paper we only discuss the word-cloud part of the design. Further details are in [21].

B. The Algorithm

Notation: We use C to denote a *corpus*, a collection of documents; D to denote a single document, and t to denote a term.

1) The Counting of Terms

First, the terms occurring in every proceeding are tokenized and counted. Terms can both be individual words or bigrams. Terms are stemmed by the version of the Snowball stemming algorithm [18] corresponding to the language of the document. In this way 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 the stem in that specific proceeding.

Kaptein, Hiemstra & Kamps [2] 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.

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 [15] and applied to word clouds before in [2] and [19]. As we slightly extended their approach for this system, we will now describe their approach and which modifications were made.

a) 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. The model estimates P(t|D), the probability of a term 't' occurring in document 'D', given 'D'.

E-STEP:

$$e_t = \mathrm{tf}(t \text{ in } D) \cdot \frac{\lambda P(t|D)}{(1-\lambda)P(t|C)+\lambda P(t|D)} \quad (2)$$

M-STEP:

$$P(t|D) = \frac{e_t}{\sum_t e_t} \quad (3)$$

In this EM-algorithm, in every iteration P(t|C) and $\mathrm{tf}(t \text{ in } D)$ remain the same. Here $\mathrm{tf}(t \text{ in } D)$ is the number of times term t occurs in document D and P(t|C) is the probability of a term in the Corpus C. P(t|C) is estimated with a maximum likelihood estimate. The term P(t|D) is re-estimated every iteration until convergence. Before the first iteration it is however simply estimated to be the inverse of the number of unique terms in the document (making the probabilities of all the terms occurring in the document summing up to 1).

The algorithm is then repeated a fixed number of steps (100 in our implementation) until (near) convergence. After every M-step, terms with a P(t|D) below 0.001 are considered non-discriminating and removed. This last step being crucial, as its goal is to remove 'non-discriminating' terms.

The goal of the complete algorithm is 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 consisting of the term t. In other words: when someone is looking for document D, how likely is it that she will use t as 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.

b) Extended approach

BIGRAM MODEL

In our extended approach, words and bigrams were modeled separately: both are stemmed, but counted apart. This was done because an extra λ2-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 λ2-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 t now becomes:

E-STEP:

$$e_t = tf(t\ in\ D) \cdot \frac{\lambda_1 P(t|D)}{(1-\lambda_1-\lambda_2)P(t|C)+\lambda_1 P(t|D)+\lambda_2 P_{bigram}(t|D)} \quad (4)$$

M-STEP:

$$P(t|D) = \frac{e_t}{\sum_t e_t} \quad (5)$$

And:

$$P_{bigram}(t|D) = \sum_{b\ in\ D} P(b|D)\,, where\ t \in bigram\ b \quad (6)$$

where Pbigram(t|D) represents the parsimonious probability that the unigram t 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 2 and 3) that contain the specific unigram.

An important note to make is that when λ2 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 λ2-parameter without doing harm to the success of the original model.

If λ2 was increased, unigrams that already occurred in a relevant bigram occurring in the word cloud got a high value of λ2Pbigram(t|D) and thereby a low P(t|D) and would disappear from the word cloud. However, the effect of the λ2-parameter on the occurrence of these superfluous unigrams differed across corpora of different size and source and was influenced by the choice of λ1 and the bigram bonus, making it hard to find the right λ2-parameter setting. More work on the exact nature of this is needed.

BIGRAM BONUS

Kaptein, Hiemstra, & Kamps [2] showed that bigrams can be more meaningful than unigrams and semantically discriminate a document from its corpus. To further test this hypothesis for our system, it has the possibility to give bigram terms a bigram bonus: a factor with which we multiply the (parsimonious) values indicating the discriminating quality of bigrams.

3) From model to clouds

When this system would be scaled up, it would be computationally very inefficient to recalculate the parsimonious language models every time the web server receives a page request. That is why their results are calculated in advance and stored in a database

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 is 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) 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. As the system was also applied on proceedings of our faculty's student council, the influence of the nature of the corpus could be tested. It turned out that the corpus itself and the length and properties of the document that is compared to it have an (undesirably) important effect on which parameters seem most fit.

Another difficulty is created by the fact that a corpus always has a lot more unique bigrams (in the order of quadratic) than words. We will discuss our findings in choosing the right parameters.

- **λ_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 an 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 can 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 our system we chose 50 as a good estimate.

- **Bigram bonus proceedings:** Especially a 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. [2] 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 worked better.

V. Empirical Evaluation of Parsimonious Language Models

A. 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?

B. Method

A group of 12 students, all members of our faculty student council, have indicated their preference for clouds constructed from 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. 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 $\lambda 1$ of 0.01, a $\lambda 2$ of 0.001, no bigram bonuses and a minimum number of unique 50 stems per topic. The clouds contained 25 terms.

C. 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).

TABLE I. Results of User Study, Percentage of Preference and SD

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%)

D. Conclusions

Parsimonious language models turned out to 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'. These are indeed words that parsimonious models filter out easily and 'for free'. Also, they often enjoyed browsing through the word clouds and suddenly remembering certain parts of meetings they attended.

VI. Discussion and Future Work

In this study we applied and extended the idea of [2] 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 this work of Kaptein, Hiemstra and Kamps 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 validated the findings of [1] in a representative user study with real users which were familiar with the documents that were being summarized.

Future work should foremost be done on setting and a better understanding of the parameters. Our study 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 $\lambda 2$-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.

REFERENCES

[1] M. Marx and A. Schuth, "DutchParl A corpus of parliamentary proceedings in Dutch", Proceedings LREC 2010, pp. 3670-3677

[2] R. Kaptein, D. Hiemstra, and J. Kamps, "How Different are Language Models and Word Clouds?," in Advances in Information Retrieval: 32nd European Conference on IR Research (ECIR 2010), Milton Keynes, 2010, p. toaear.

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

[4] A. W. Rivadeneira, D. M. Gruen, M. J. Muller, and D. R. Millen, "Getting our head in the clouds: toward evaluation studies of tagclouds.," in Proceedings CHI 2007, New York, 2007, pp. 995-998.

[5] (2011) Political Mashup. [Online]. http://www.politicalmashup.nl/

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

[7] C. Aalberts, Aantrekkelijke Politiek?. Amsterdam: Spinhuis / Maklu, Het , 2006.

[8] C. Aalberts, "Politieke betrokkenheid en politieke sensitiviteit onder jongeren," in Workshop Kwaliteit van het leven en politieke attitudes, Antwerpen, 2004.

[9] I. Costera Meijer, De toekomst van het nieuws. Amsterdam: Otto Cramwinkel Uitgever, 2006.

[10] M. Gebuis and A. van Hoof, "De dogma's van het nieuws voorbij. Het effect van postmoderne nieuwsmedia op waardering, blootstelling en

politieke betrokkenheid bij jongeren," in Etmaal van de Communicatiewetenschap 2010, Gent, 2010.

[11] M. Marx and T. Gielissen, "Digital Weight Watching: Reconstruction of scanned documents," in Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data, Barcelona, Spain, 2009, pp. 25-31.

[12] J. Feinberg. (2010) Wordle.net: Beautiful Word Clouds. [Online]. http://www.wordle.net

[13] IBM. (2010) Many Eyes: Tag Cloud. [Online]. http://manyeyes.alphaworks.ibm.com/manyeyes/page/Tag_Cloud.html

[14] M. F. Porter, "Program, 14(3)," An algorithm for suffix stripping, p. 130−137, 1980.

[15] D. Hiemstra, S. Robertson, and H. Zaragoza, "Parsimonious language models for information retrieval," in Proceedings SIGIR, New York, 2004, pp. 178-185.

[16] European Parliament, deparment The Netherlands. (2010) Nederlanders en Europa. [Online]. http://www.europa-nu.nl/id/vh93qqnk8atd/nederland_over_europa

[17] eXist-db. (2010) eXist-db Open Source Native XML Database. [Online]. http://exist.sourceforge.net/

[18] M. Porter. (2010) Snowball Homepage. [Online]. http://snowball.tartarus.org/index.php

[19] R. Kaptein and M. Marx, "Focused Retrieval and Result Aggregation with Political Data," Information Retrieval, 13(5) p. 412-433, 2010.

[20] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University Press. 2008.

[21] Gilles den Hollander, “The EU parliament in clouds”, Bsc thesis, University of Amsterdam, 2010. http://politicalmashup.nl/2010/12/scriptie-gilles-de-hollander/