How Different are Language Models and Word Clouds?

Rianne Kaptein[1] Djoerd Hiemstra[2] Jaap Kamps[1,3]

[1] Archives and Information Studies, Faculty of Humanities, University of Amsterdam
[2] Database Group, University of Twente
[3] ISLA, Informatics Institute, University of Amsterdam

ABSTRACT

Word clouds are a summarised representation of a document's text, similar to tag clouds which summarise the tags assigned to documents. Word clouds are similar to language models in the sense that they represent a document by its word distribution. In this paper[1] we investigate the differences between word cloud and language modelling approaches, and specifically whether effective language modelling techniques also improve word clouds. We evaluate the quality of the language model and the resulting word clouds using a system evaluation test bed, and a user study. Our experiments show that different language modelling techniques can be applied to improve a standard word cloud that uses a TF weighting scheme in combination with stopword removal. Including bigrams in the word clouds and a parsimonious term weighting scheme are the most effective in both the system evaluation and the user study.

1. INTRODUCTION

This paper investigates the connections between tag or word clouds popularised by Flickr and other social web sites, and the language models as used in IR. The new generation of the Internet, the social Web, allows users to do more than just retrieve information and engages users to be active. Users can now add tags to categorise web resources and retrieve their own previously categorised information. By sharing these tags among all users large amounts of resources can be tagged and categorised. These generated user tags can be visualised in a tag cloud where the importance of a term is represented by font size or colour. Of course, the majority of documents on the web are not tagged by users. An alternative to clouds based on user-assigned tags, is to generate clouds automatically by using statistical techniques on the document contents, so-called 'word clouds'. Figure 1 shows a word cloud summarising 10 documents. Our main research question is: do words extracted by language modelling techniques correspond to the words that users like to see in word clouds?

2. EXPERIMENTS

Since there is no standard evaluation method for word clouds, we created our own experimental test bed. Our experiments comprise of two parts, a system evaluation and a user study. For both experiments we use query topics from the 2008 TREC Relevance Feedback track. The system evaluation consists of two parts, first we test if adding the word cloud as a whole to the original query leads to improvements in retrieval performance. Secondly, for each topic we generate 25 queries where in each query one word from the word cloud is added to the original query. For each query we measure the difference in performance caused by adding the expansion term to the original query, words are considered relevant if adding the word leads to an improvement in retrieval performance, words are considered acceptable if there is no large decrease (more than 25%) in retrieval results. In the user study test persons rank different groups of word clouds. The 13 test persons consisted of 4 females and 9 males with ages ranging from 26 to 44 and were recruited at the university.

Table 1: Effectiveness of unigrams and bigrams

Approach	MAP	P10	% Rel. words	% Acc. words
Unigrams	0.2575	0.5097	**35**	**73**
Mixed	**0.2706** $^{-}$	**0.5226** $^{-}$	31	71
Bigrams	0.2016°	0.4387 $^{-}$	25	71

Table 2: Effectiveness of term weighting approaches

Approach	MAP	P10	% Rel. words	% Acc. words
TF	0.2575	0.5097	**35**	**73**
TFIDF	0.1265$^{\bullet}$	0.3839°	22	67
Pars.	**0.2759**$^{\circ}$	**0.5323** $^{-}$	31	68

Clouds from Pseudo Relevant and Relevant Results

First, we compare a TF cloud made from 10 pseudo-relevant documents to a cloud of 100 relevant documents. We make this comparison to get some insights on the question whether there is a mismatch between words that improve retrieval performance, and words that users like to see in a word cloud. Our standard word cloud (shown in Figure 1) uses pseudo-relevant results. The cloud in Fig. 2 is based on 100 pages judged as relevant.

When we look at the system evaluation the relevant documents lead to better performance than the pseudo-relevant documents. The test persons in our user study however clearly prefer the clouds based on 10 pseudo-relevant documents: 66 times the pseudo-relevant cloud is preferred, 36 times the relevant cloud, and in 27 cases there is no preference (significant at 95% using a two-tailed sign-test). There seem to be three groups of words that often contribute positively to retrieval results, but are probably not appreciated by test persons: numbers, general and frequently occurring words which do not seem specific to the query topic e.g. 'year' or 'up', words that test persons don't know like abbreviations or technical terms .

Non-Stemmed and Conflated Stemmed Clouds

We look at the impact of stemming by generating conflated stemmed

[1]This paper is a compressed version of Kaptein, R., Hiemstra, D., and Kamps, J. (2010). How different are language models and word clouds? In Advances in Information Retrieval: 32nd European Conference on IR Research (ECIR 2010), volume 5993 of LNCS, pages 556-568. Springer.

DIR'2011, February 4, 2011, Amsterdam.

Figure 1: Word cloud from 10 results for the topic "diamond smuggling"

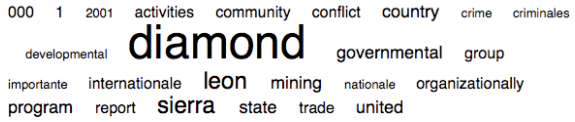

Figure 2: Word cloud from 100 relevant results

clouds. To stem, we use the most common English stemming algorithm, the Porter stemmer [2]. To visualize terms in a word cloud, Porter word stems are not a good option. A requirement for the word clouds is to visualize correct English words, and not stems of words which are not clear to the user, therefore in our conflated word clouds, word stems are replaced by the most frequently occurring word in the collection that can be reduced to that word stem. The effect of stemming is only evaluated in the user study. Looking at pairwise preferences, we see that there is no significant preference for the conflated cloud or the non-stemmed cloud. Often the difference between the clouds is so small that it is not noticed by test persons.

Bigrams

For users, bigrams are often easier to interpret than single words, because a little more context is provided. We have created two models that incorporate bigrams, a mixed model that contains a mix of unigrams and bigrams, and a bigram model that consists solely of bigrams. For the user study we placed bigrams between quotes to make them more visible as can be seen in Figure 3. In Table 1 the system evaluation results are shown. For query expansion, the model that uses a mix of unigrams and bigrams performs best. Using only bigrams leads to a significant decrease in retrieval results compared to using only unigrams. Looking at the percentages of relevant and acceptable words, the unigram model produces the most relevant words. The mixed model performs almost as good as the unigram model.

In the user study, the clouds with mixed unigrams and bigrams and the clouds with only bigrams are selected most often as the best cloud. There is no significant difference in preference between mixed unigrams and bigrams, and only bigrams. Users do indeed like to see bigrams, but for some queries the cloud with only bigrams contains too many meaningless bigrams such as 'http www'. An advantage of the mixed cloud is that the number of bigrams in the cloud is flexible. When bigrams occur often in a document, also many will be included in the word cloud.

Term Weighting

Besides the standard TF weighting we investigate two other variants of language models to weigh terms, the TFIDF model and the parsimonious model. Before weighting terms we always remove an extensive stopword list consisting of 571 common English words. In the TFIDF algorithm, the text frequency (TF) is now multiplied by the inverse document frequency (IDF).

The third variant of our term weighting scheme is a parsimonious model [1]. The parsimonious language model concentrates the probability mass on fewer words than a standard language model.

Figure 3: Word cloud of 10 results with mixed unigrams and bigrams

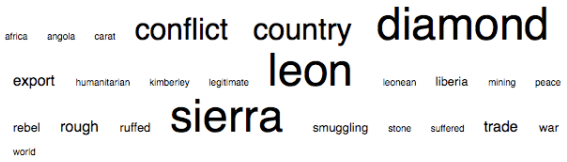

Figure 4: Word cloud of 10 results with parsimonious term weighting.

In Figure 4 the parsimonious word cloud of our example topic is shown. Table 2 shows the system evaluation results for the different term weighting schemes.

The parsimonious model performs best on both early and average precision. The TFIDF model performs significantly worse than the TF and the parsimonious model. Our simplest model, the TF model, actually produces the highest number of relevant and acceptable words. The weighting scheme of the parsimonious model is clearly more effective than the TF model though, since for query expansion where weights were considered the parsimonious model performed better than the TF model.

In the user study the parsimonious model is preferred more often than the TF model, and both the parsimonious and the TF model are significantly more often preferred over the TFIDF model. The parsimonious model contains more specific and less frequently occurring words than the TF model.

3. CONCLUSION

This paper investigated the connections between word clouds and the language models as used in IR. We have investigated how we can create word clouds from documents and use language modelling techniques which are more advanced than only frequency counting and stopword removal. We find that different language modelling techniques can indeed be applied to create better word clouds. Including bigrams in the word clouds and a parsimonious term weighting scheme are the most effective improvements. We find there is some discrepancy between good words for query expansion selected by language modelling techniques, and words liked by users. This will be a problem when a word cloud is used for suggestion of query expansion terms. The problem can be partly solved by using a parsimonious weighting scheme which selects more specific and informative words than a TF model, but also achieves good results from a system point of view.

REFERENCES

[1] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *Proceedings SIGIR'04*, pages 178–185. ACM Press, New York NY, 2004.

[2] M. Porter. An algorithm for suffix stripping. *Program*, 14(3): 130–137, 1980.