Word Clouds of Multiple Search Results

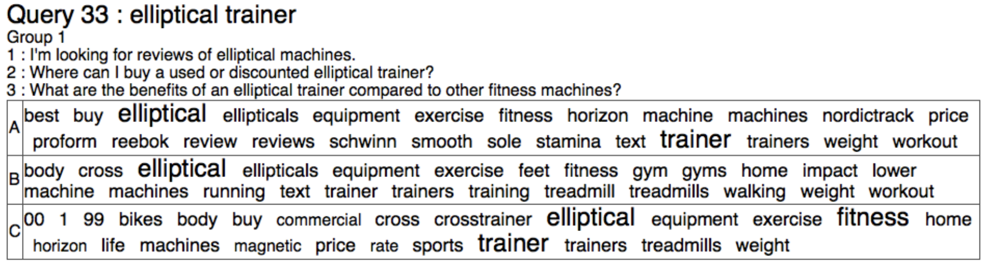
Rianne Kaptein[1] and Jaap Kamps[1,2]

[1] Archives and Information Studies, University of Amsterdam, the Netherlands
[2] ISLA, Informatics Institute, University of Amsterdam, the Netherlands

Abstract. Search engine result pages (SERPs) are known as the most expensive real estate on the planet. Most queries yield millions of organic search results, yet searchers seldom look beyond the first handful of results. To make things worse, different searchers with different query intents may issue the exact same query. An alternative to showing individual web pages summarized by snippets is to represent whole group of results. In this paper we investigate if we can use word clouds to summarize groups of documents, e.g. to give a preview of the next SERP, or clusters of topically related documents. We experiment with three word cloud generation methods (full-text, query biased and anchor text based clouds) and evaluate them in a user study. Our findings are: First, biasing the cloud towards the query does not lead to test persons better distinguishing relevance and topic of the search results, but test persons prefer them because differences between the clouds are emphasized. Second, anchor text clouds are to be preferred over full-text clouds. Anchor text contains less noisy words than the full text of documents. Third, we obtain moderately positive results on the relation between the selected world clouds and the underlying search results: there is exact correspondence in 70% of the subtopic matching judgments and in 60% of the relevance assessment judgments. Our initial experiments open up new possibilities to have SERPs reflect a far larger number of results by using word clouds to summarize groups of search results.

1 Introduction

In this paper we investigate the use of word clouds to summarize multiple search results. We study how well users can identify the relevancy and the topic of search results by looking only at the word clouds. Search results can contain thousands or millions of potentially relevant documents. In the common search paradigm of today, you go through each search result one by one, using a search result snippet to determine if you want to look at a document or not. We want to explore an opportunity to summarize multiple search results which can save the users time by not having to go over every single search result. Documents are grouped by two dimensions. First of all, we summarize complete SERPs containing documents returned in response to a query. Our goal is to discover whether a summary of a SERP can be used to determine the relevancy of the search results on that page. If that is the case, such a summary can for example be placed at the bottom of a SERP so the user can determine if he wants to look at the next result page, or take another action such as rephrasing the query.


Query 33 : elliptical trainer

Group 1

1 : I'm looking for reviews of elliptical machines.

2 : Where can I buy a used or discounted elliptical trainer?

3 : What are the benefits of an elliptical trainer compared to other fitness machines?

A	best buy elliptical ellipticals equipment exercise fitness horizon machine machines nordictrack price proform reebok review reviews schwinn smooth sole stamina text trainer trainers weight workout
B	body cross elliptical ellipticals equipment exercise feet fitness gym gyms home impact lower machine machines running text trainer trainers training treadmill treadmills walking weight workout
C	00 1 99 bikes body buy commercial cross crosstrainer elliptical equipment exercise fitness home horizon life machines magnetic price rate sports trainer trainers treadmills weight

Fig. 1. Full-text clouds for the query 'Elliptical Trainer" of the subtopic matching task

Secondly, documents are grouped by subtopic of the search request. Search results are usually documents related to the same topic, that is the topic of the search request. However, a query can be related to different user needs where a distinction can be made between ambiguous and faceted queries. Ambiguous queries are those that have multiple distinct interpretations, and most likely a user interested in one interpretation would not be interested in the others. Faceted queries are underspecified queries with different relevant aspects, and a user interested in one aspect may still be interested in other aspects [3]. In this paper facets and interpretations of ambiguous queries are both considered as subtopics of the query.

Clustering search results into subtopics of the query can organise the huge amount of search results. Efficiently summarising these clusters through the use of a word cloud can help the users select the right cluster for their search request. Examples of a word cloud can be found in Figure 1. These clouds are generated for subtopics of the query 'elliptical trainer'[3]. For each of the three subtopics a word cloud is generated from documents relevant to those subtopics.

Tag and word clouds are being explored for multiple functions, mainly on the social Web. Tag clouds summarize the tags assigned by users to documents, whereas word clouds can summarize documents without user assigned tags. Since there is no need for a manual effort to generate word clouds, there is a much larger potential of document sets where word clouds can be helpful. Terms in a tag cloud usually link to a collection of documents that are associated with that tag.

An advantage of word clouds is that there is no need for high quality, grammatically correct text in the documents. Using word clouds we can make summaries of web results like twitter streams, blogs, or transcribed video. Since the transcriptions usually still contain a considerable number of errors they are not suitable for snippet generation for examples. Word clouds are a good alternative, also because repeatedly occurring words have a higher chance of getting recognized [18]. Also we can make use of anchor text, which is a source of information that is used to rank search results, but which is not usually visible to the user. The anchor text representation of a web document is a collection of all the text which is used on or around the links to a document. Again, anchor text does not consist of grammatically correct sentences, but it does contain a lot of repetition, which is advantageous for the generation of word clouds.

In this paper we want to answer the following main research question:

[3] This is topic 33 of the 2009 TREC Web track [3]

How can we use word clouds to summarize multiple search results to convey the topic and relevance of these search results?

In the context of search, we want to investigate the following issues. The snippets used in modern web search are query biased, and are proven to be better than static document summaries. We want to examine if the same is true for word clouds, hence our first research question is:

Are query biased word clouds to be preferred over static word clouds?

Besides the text on a web page, web pages can be associated with anchor text, i.e. the text on or around links on web pages linking to a web page. This anchor text is used in many search algorithms. Our second research question is:

Is anchor text a suitable source of information to generate word clouds?

The remainder of this paper is organized as follows. In the next section we discuss related work. Section 3 describes the models we use to generate the word clouds. In section 4 we evaluate the word clouds by means of a user study. Finally, in section 5 we draw our conclusions.

2 Related Work

In this section we discuss related work on snippets and alternative search result presentations, cluster labeling, keyphrase extraction and tag clouds. Many papers on search result summaries focuses on single documents, where the snippet is the most common form of single document summarization. It has been shown that query biased snippets are to be preferred over static document summaries consisting of the first few sentences of the document [17]. Query biased summaries assist users in performing relevance judgements more accurately and quickly, and they alleviate the users' need to refer to the full text of the documents.

An alternative to the traditional web search result page layout is investigated in [21]. Sentences that highly match the searcher's query and the use of implicit evidence are examined, to encourage users to interact more with the results, and to view results that occur after the first page of 10 results.

Another notable search application with an alternative search interface is PubCloud. PubCloud is an application that queries PubMed for scientific abstracts and summarizes the responses with a tag cloud interface [11]. A stopword list is used to remove common words, and a Porter stemmer is applied. Colours are used to represent recency, and font size represents frequency. Mousing over a tag displays a list of words that share the same prefix and a hyperlink links to the set of PubMed abstracts containing the tag.

Related research is done in the field of cluster labeling and the extraction of keywords from documents. Similar to our word cloud generation algorithms, these techniques extract words that describe (clusters of) documents best.

Pirolli et al. [12] present a cluster-based browsing technique for large text collections. Clusters of documents are generated using a fast clustering technique based on pairwise document similarity. Similar documents are placed into the same cluster. Recursively clustering a document collection produces a cluster hierarchy. Document clusters are summarized by topical words, the most frequently occurring words in a cluster, and typical titles, the words with the highest similarity to a centroid of the cluster. Participants in a user study were asked to rate the precision of each cluster encountered. It was shown that summarization by keywords is indeed suitable to convey the relevance of document clusters.

The goal of cluster labeling is to find the single best label for a cluster, i.e. the label equal to a manually assigned label, these algorithms generate a ranking of possible labels, and success is measured at certain cut-offs or through a Mean Reciprocal Rank. Manually assigned category labels are extracted for example from the internet directory DMOZ such as is done in [2]. The set of terms that maximizes the Jensen-Shannon Divergence distance between the cluster and the collection is considered as cluster label. Wikipedia is used as an external source from which candidate cluster labels can be extracted. Instead of the text of the documents Glover et al. [5] use the extended anchor text of web pages to generate cluster labels.

In the machine learning community, a similar task is keyphrase extraction. Here, the task is seen as a classification task, i.e. the problem is to correctly classify a phrase into the classes ‘keyphrase’ and ‘not-keyphrase’ [4]. A keyphrase can contain up to three or sometimes five words. While information retrieval approaches usually consider documents as “bags-of-words”, except for term dependencies, some keyphrase extraction techniques also take into account the absolute position of words in a document. The Kea keyphrase extraction algorithm of Frank et al. [4] uses as a feature the distance of a phrase from the beginning of a document, which is calculated as the number of words that precede its first appearance, divided by the number of words in the document. The basic feature of this and the following algorithms is however a frequency measure, i.e. TF*IDF (Term Frequency*Inverse Document Frequency). Turney [19] extends the Kea algorithm by adding a coherence feature set that estimates the semantic relatedness of candidate keyphrases aiming to produce a more coherent set of keyphrases. Song et al. [15] use also a feature ‘distance from first occurrence’. In addition, part of speech tags are used as features. The extracted keyphrases are used for query expansion, leading to improvements on TREC ad hoc sets and the MEDLINE dataset.

While for snippets it is clear that query biased snippets are better than static summaries, cluster labels are usually static and not query dependent. Many experiments use web pages as their document set, but the extracted labels or keyphrases are not evaluated in the context of a query which is the purpose of this study.

Most works concerning tag and word clouds address the effects of visual features such as font size, font weight, colour and word placement [1, 7, 14]. General conclusions are that font size and font weight are considered the most important visual properties. Colour draws the attention of users, but the meaning of colours is not always obvious. The position of words is important, since words in the top of the tag cloud attract more attention.

In previous work we studied the similarities and differences between language models and word clouds [10]. Word clouds generated by different approaches are evaluated by a user study and a system evaluation. Two improvements over a word cloud based on text frequency and the removal of stopwords are found. Applying a parsimonious term weighting scheme filters out not only common stopwords, but also corpus specific stopwords and boosts the probabilities of the most characteristic words. Secondly, the inclusion of bigrams into the word clouds is appreciated by our test persons. Single terms are sometimes hard to understand when they are out of context, while the meaning of bigrams stays clear even when the original context of the text is missing. In this work we take more contextual information into account, namely the query that was used to generate the search results and the anchor text of the search results.

Most tag clouds on the Web are generated using simple frequency counting techniques. While this works well for user-assigned tags, we need more sophisticated models to generate word clouds from documents. These models will be discussed in the next section.

3 Word Cloud Generation

We generate word clouds using the language modeling approach. We choose this approach because it is conceptually simple. The approach is based on the assumption that users have some sense of the frequency of words and which words distinguish documents from others in the collection [13]. As a pre-processing step we strip the HTML code from the web pages to extract the textual contents. We use three models to generate the word clouds.

3.1 Full-Text Clouds

In previous work we have shown that the parsimonious language model is a suitable model to generate word clouds [9]. The parsimonious language model [8] is an extension to the standard language model based on maximum likelihood estimation, and is created using an Expectation-Maximization algorithm. Maximum likelihood estimation is used to make an initial estimate of the probabilities of words occurring in the document.

$$P_{mle}(t_i|D) = \frac{tf(t_i, D)}{\sum_t tf(t, D)} \tag{1}$$

where D is document, and $tf(t, D)$ is the text frequency, i.e. the number of occurrences of term t in D. Subsequently, parsimonious probabilities are estimated using *Expectation-Maximisation*:

$$\begin{aligned} &\text{E-step: } e_t = tf(t, D) \cdot \frac{(1-\lambda)P(t|D)}{(1-\lambda)P(t|D) + \lambda P(t|C)} \\ &\text{M-step: } P_{pars}(t|D) = \frac{e_t}{\sum_t e_t}, \text{ i.e. normalize the model} \end{aligned} \tag{2}$$

where C is the background collection model. In the initial E-step, maximum likelihood estimates are used for $P(t|D)$. Common values for the smoothing parameter λ are 0.9 or 0.99. We see that when $\lambda = 0.9$, the word clouds contain a lot of very general words and many stopwords. When $\lambda = 0.99$ the word clouds contain more informative words, and therefore in the rest of this work we set λ to 0.99. In the M-step the words that receive a probability below our threshold of 0.0001 are removed from the model. This threshold parameter determines how many words are kept in the model and does not affect the most probable words, which are used for the word clouds. In the next iteration the probabilities of the remaining words are again normalized. The iteration process stops after a fixed number of iterations.

Instead of generating word clouds for single documents, we create word clouds for sets of documents. We want to increase the scores of words which occur in multiple documents. This is incorporated in the parsimonious model as follows:

$$P_{mle}(t_i|D_1,\ldots,D_n) = \frac{\sum_{i=1}^{n} tf(t,D_i)}{\sum_{i=1}^{n}\sum_{t} tf(t,D_i)} \tag{3}$$

The initial maximum likelihood estimation is now calculated over all documents in the document set $D_1,\ldots,D_n$. This estimation is similar to treating all documents as one single aggregated document. The E-step becomes:

$$e_t = \sum_{i=1}^{n} tf(t,D_i) * df(t,D_i,\ldots,D_n) \cdot \frac{(1-\lambda)P(t|D_1,\ldots,D_n)}{(1-\lambda)P(t|D_1,\ldots,D_n)+\lambda P(t|C)} \tag{4}$$

In the E-step also everything is calculated over the set of documents now. Moreover, to reward words occurring in multiple documents we multiply the term frequencies tf by the document frequencies df, the number of documents in the set in which the term occurs, i.e., terms occurring in multiple documents are favoured. The M-step remains the same.

Besides single terms, multi-gram terms are suitable candidates for inclusion in word clouds. Most social websites also allow for multi-term tags. Our n-gram word clouds are generated using an extension of the bigram language model presented in [16]. We extend the model to a parsimonious version, and to consider n-grams. Our n-gram language model uses only ordered sets of terms. The model based on term frequencies then looks as follows:

$$\begin{aligned} &P_{mle}(t_j,\ldots,t_m|D_i,\ldots,D_n) \\ &\quad= \frac{\sum_{i=1}^{n} tf(t_j,\ldots,t_m,D_i)}{\text{argmin}_{j=1,\ldots,m}\sum_{i=1}^{n} tf(t_j,D_i)} * \frac{df(t_j,\ldots,t_m,D_i,\ldots,D_n)}{n} \end{aligned} \tag{5}$$

The parsimonious version of this model takes into account the background collection to determine which n-grams distinguish the document from the background collection. To promote the inclusion of terms consisting of multiple words, in the E-step of the parsimonious model we multiply e_t by the length of the n-gram. Unfortunately, we do not have the background frequencies of all n-grams in the collection. To estimate the background probability $P(t_j,\ldots,t_m|C)$ in the parsimonious model we therefore use a linear interpolation of the smallest probability of the terms in the n-gram occurring in the document, and the term frequency of this term in the background collection.

Another factor we have to consider when creating a word cloud is overlapping terms. The word cloud as a whole should represent the words that together have the greatest possible probability mass of occurrence. That means we do not want to show

```
Create a set of n-gram terms ranked by their scores to potentially include in the cloud
while the maximum number of terms in the cloud is not reached do
    Add the highest ranked term to the cloud
    Subtract the score of the term from the score of its head and tail
    if The head or tail of the term is already in the cloud then
        Remove it from the cloud, and insert it to the set of potential terms again
    end if
end while
```

Fig. 2. Pseudo-code for constructing a n-gram cloud from a set of ranked terms

single terms that are also part of a multi-gram term, unless this single term occurs with a certain probability without being part of the multi-gram term. We use the algorithm depicted in Fig. 2 to determine which words to include in the cloud. The head of a n-gram term is the term without the last word, likewise the tail is the term without the first word.

To determine the size of a term in the clouds we use a log-scale to bucket the terms into four different font sizes according to their probabilities of occurrence.

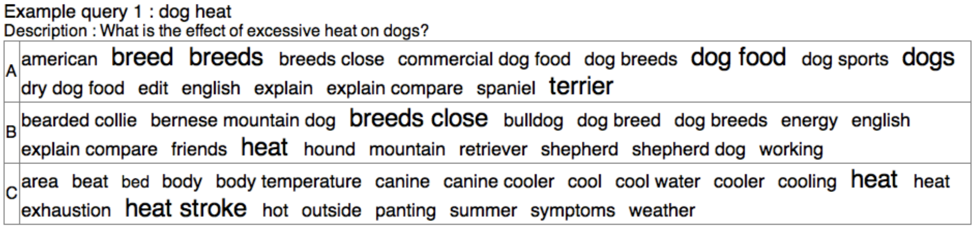
3.2 Query Biased Clouds

In the parsimonious model the background collection C is used to determine what are common words in all documents to determine what words distinguish a certain document from the background collection. In our case where documents are returned for the same search request, it is likely that these documents will be similar to each other. All of them will for example contain the query words. Since we want to emphasize the differences between the groups of search results, we should use a smaller and more focused background collection. So in addition to the background collection consisting of the complete document collection, we use a topic specific background collection. For the documents grouped by relevance, the topic specific background collection consists of the top 1,000 retrieved documents of a search topic. For the documents grouped by subtopic of the search request, the topic-specific background collection consists of all documents retrieved for any subtopic. Using background models on different levels of generality helps to exclude non-informative words.

We estimate a mixed model with parsimonious probabilities of a word given two background collections as follows:

$$\begin{aligned} &\text{E-step: } e_t = tf(t,D) \cdot \frac{(1-\lambda-\mu)P(t|D)}{(1-\lambda-\mu)P(t|D) + \lambda P(t|C_1) + \mu P(t|C_2)} \\ &\text{M-step: } P_{pars}(t|D) = \frac{e_t}{\sum_t e_t}, \text{ i.e. normalize the model} \end{aligned} \qquad (6)$$

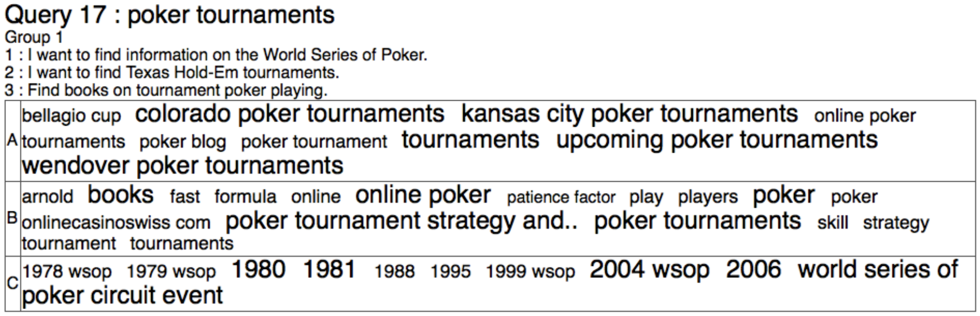
There are two background models: C_1 and C_2. C_1 is the model based on the complete corpus. C_2 is the topic specific model. The weight of the background models is



Example query 1 : dog heat
Description : What is the effect of excessive heat on dogs?

A	american breed breeds breeds close commercial dog food dog breeds dog food dog sports dogs dry dog food edit english explain explain compare spaniel terrier
B	bearded collie bernese mountain dog breeds close bulldog dog breed dog breeds energy english explain compare friends heat hound mountain retriever shepherd shepherd dog working
C	area beat bed body body temperature canine canine cooler cool cool water cooler cooling heat heat exhaustion heat stroke hot outside panting summer symptoms weather

Fig. 3. Query biased clouds for the query 'Dog Heat' of the relevance assessment task



Query 17 : poker tournaments
Group 1
1 : I want to find information on the World Series of Poker.
2 : I want to find Texas Hold-Em tournaments.
3 : Find books on tournament poker playing.

A	bellagio cup colorado poker tournaments kansas city poker tournaments online poker tournaments poker blog poker tournament tournaments upcoming poker tournaments wendover poker tournaments
B	arnold books fast formula online online poker patience factor play players poker poker onlinecasinoswiss com poker tournament strategy and.. poker tournaments skill strategy tournament tournaments
C	1978 wsop 1979 wsop 1980 1981 1988 1995 1999 wsop 2004 wsop 2006 world series of poker circuit event

Fig. 4. Anchor text clouds for the query 'Poker tournaments' of the subtopic matching task

determined by two parameters, λ and μ. We keep the total weight of the background models equal at 0.99, so we choose for λ and μ a value of 0.495.

Our standard model uses the full text of documents to generate word clouds. In addition to using a focused background collection, we focus on the text around the query terms to generate query biased clouds. The surrogate documents used to generate query biased clouds contain only terms that occur around the query words. In our experiments all terms within a proximity of 15 terms to any of the query terms is included.

3.3 Anchor Text Clouds

So far, we used the document text to generate word clouds. On the web however, there is another important source of information that can be used to summarize documents: the anchor text. When people link to a page, usually there is some informative text contained in the link and the text around the link.

The distribution of anchor text terms greatly differs from the distribution of full-text terms. Some webpages do not have any anchor text, while others have large amounts of (repetitive) anchor text. As a consequence we can not use the same language models to model full text and anchor text. Anchor texts are usually short and coherent. We therefore treat each incoming anchor text as one term, no matter how many words it contains. For each document, we only keep the most frequently occurring anchor text term. The terms are cut off at a length of 35, which only affects a small number of terms. Maximum likelihood estimation is used to estimate the probability of an anchor text term occurring, dividing the number of occurrences of the anchor text by the total number of anchor text terms in the document set. When after adding all the anchor text terms to the word cloud the maximum number of terms in the cloud is not reached, the anchor text cloud is supplemented with the highest ranked terms from the document's full text.

4 Experiments

We conduct a user study to evaluate our word cloud generation models. After describing the set-up, results are given and analyzed.

4.1 Experimental Set-Up

To evaluate the quality of our word clouds we perform a user study consisting of two tasks. The set-up of the user study is as follows. Participants are recruited by e-mail. The user study is performed online and starts with an explanation of the task, including some examples and a training task. A short pre-experiment questionnaire follows, before the experiment starts with the subtopic matching task, which consists of 10 queries. Three versions of the study are generated, which together cover 30 queries for each part of the study. A version is randomly assigned when a test person starts the study.

For each query two groups of clouds have to be matched to particular subtopics. The three methods described in the previous section are used to generate the groups of word clouds: Full-Text (FT), Query biased (QB), and Anchor text (AN). The two groups of clouds are generated using two out of the three word cloud generation methods, which are selected using a rotation scheme. The test persons do not know which word cloud generation methods are used. Besides the matching task, the test persons also assign a preference for one of the two groups. The second part of the study is the relevance assessment task, which consists of 10 queries with two groups of clouds. Again for each query two out of the three word cloud generation methods are selected using a rotation scheme. Finally, a post-experiment questionnaire finishes the user study.

We use different sets of queries for each pair of word cloud generation methods allowing for pairwise comparison. Since the query effect is large due to differences in the quality of retrieved documents, we cannot compare all three methods on the same grounds.

Task 1: Subtopic Matching: When queries are ambiguous or multi faceted, can the word cloud be used to identify the clusters? To evaluate the disambiguation potential of word clouds we let test persons perform a matching task. Given a query, and a number of subtopics of this query, test persons have to match the subtopics to the corresponding word clouds. An example topic for this task can be found in Figure 1.

Topics are created as follows. We use topics from the diversity task in the TREC 2009 Web track [3]. Topics for the diversity task were created from the logs of a commercial search engine. Given a target query, groups of related queries using co-clicks and other information were extracted and analysed to identify clusters of queries that highlight different aspects and interpretations of the target query. Each cluster represents a subtopic, and the clusters of related queries are manually processed into a natural language description of the subtopic, which is shown to our test persons.

The clouds in the user study are generated as follows. The relevance of documents to subtopics is judged by assessors hired by TREC. From the relevance assessments we extract relevant documents for each subtopic. A subtopic is only included if there are at least three relevant documents. Furthermore, we set a minimum of two subtopics per query topic, and a maximum of four. If there are more than four subtopics with at least three relevant documents, we randomly select four subtopics. The methods used to generate the word clouds from the selected documents are described in the previous section.

Task 2: Relevance Assessment: How well can test persons predict if results are relevant by looking at a word cloud? To evaluate this task we let test persons grade word clouds which represent a complete search result page for a particular query. These word clouds are graded by the test persons in our user study on a three-point scale (Relevant, Some relevance, Non relevant). An example topic for this task can be found in Figure 3. Three word clouds are created for each topic using 20 documents, i.e one cloud generated using only relevant documents, one cloud generated where half of the documents are relevant, and the other half of the documents are non-relevant, and one cloud generated using only non-relevant documents). In the ideal case the test person evaluates the cloud created from only relevant documents as ”Relevant”, the cloud created from non-relevant documents as ”Non relevant”, and the cloud created from the mix of relevant and non-relevant documents as ”Some relevance”.

The topics we use are taken from the ad hoc task of the TREC 2009 Web track. We use the relevance assessments of the track to identify relevant documents, and the documents from the bottom of the ranking of a standard language model run returning 1,000 results as non-relevant documents. To ensure there are differences between the relevant and the non-relevant documents, we take the documents from the bottom of the ranking of a standard language model run returning 1,000 results as non-relevant documents. There is a small chance that there are still some relevant documents in there, but most documents will not be relevant, although they will contain at least the query words.

4.2 Experimental Results

We evaluate our word cloud generation methods through the user study. This leads to the following results.

Demographics In total 21 test persons finished the complete user study. The age of the test persons ranges from 25 to 42 year, with an average age of 30. Most test persons were Dutch, but overall 11 nationalities participated. All test persons have a good command of the english language. A large part of the test persons is studying or working within the field of information retrieval or computer science. The familiarity with tag clouds is high, on average 3.8 measured on a Likert-scale, where 1 stands for ‘totally unfamiliar’

Table 1. Percentage of correct assignments on the relevance assessments task

Model	Relevant	Half	Non Relevant	All
FT	0.42	0.36	0.44	0.40
QB	0.42^{-}	0.39^{-}	0.50^{-}	0.44^{-}

Table 2. Confusion matrix of assignments on the relevance assessments task for the FT model

	Assessed as		
Generated from	Relevant	Half	Non Relevant
Relevant	*178*	180	72
Half	222	*154*	54
Non Relevant	66	174	*186*

and 5 stands for 'very familiar'. On average the test persons spent 38 minutes on the user study in total. The first task of subtopic matching took longer with an average of 19 minutes, while the second task of relevance assessments went a bit quicker with an average of 14 minutes. Since the tasks are always conducted in the same order, this could be a learning effect.

Query Biased Word Clouds We take a look at the results of both tasks in the user study (subtopic matching and relevance judgments) to answer our first research question: 'Are query biased word clouds to be preferred over static word clouds?'. The first task in the user study was to match subtopics of the search request to the word clouds. Our test persons perform the subtopic matching significantly better using the full-text model (significance measured by a 2-tailed sign-test at significance level 0.05). The full-text clouds judgments match the ground truth in 67% of all assignments, the query biased clouds match in 58% of the cases.

In the second task of the user study the test persons assess the relevance of the presented word clouds on a three-point scale. Although each group of clouds contains one cloud of each relevance level, the test persons can choose to assign the same relevance level to multiple word clouds. Since in the subtopic matching task each subtopic should be matched to only one cloud, there could be a learning effect that the test persons assign each relevance level also to only one cloud. We show the results of this task in Table 1. On the relevance assessment task the query biased model performs better than the full-text model, but the difference is not significant.

The results split according to relevance level are shown in the confusion matrices in Tables 2 and 3. We see that the clouds containing some relevance (half) match the ground truth the least often. The non relevant clouds are recognized with the highest accuracy, especially in the query biased model. When we look at the distribution of the relevance levels, it is not the case that most assignments are to 'Non relevant'. For both models the distinction between clouds generated from relevant documents, and clouds generated from a mix of relevant and non-relevant documents is the hardest to make for our test persons.

Table 3. Confusion matrix of assignments on the relevance assessments task for the QB model

	Assessed as		
Generated from	Relevant	Half	Non Relevant
Relevant	*180*	168	84
Half	222	*168*	42
Non Relevant	78	138	*216*

Table 4. Percentage of correct assignments on the relevance assessments task

Model	Relevant	Half	Non Relevant	All
FT	0.61	0.47	0.56	0.54
AN	0.62^{-}	0.50^{-}	0.63^{-}	0.59^{-}

Anchor Text Clouds We now examine our second research question 'Is anchor text a suitable source of information to generate word clouds?'. On the subtopic matching task, the anchor text model performs slightly better than the full-text model on the subtopic task, with an accuracy of 72% versus an accuracy of 68% of the full text model.

Results of the relevance assessment task can be found in Table 4. The anchor text model performs best, with almost 60% of the assignments correctly made. Again the clouds with some relevance are the hardest to recognize. The confusion matrices of both models show a pattern similar to the confusion matrices in Figure 2 and 3, and are therefore omitted here.

The inter-rater agreement for both tasks measured with Kendall's tau lies around 0.4, which means there is quite some disagreement. Besides comparing the word cloud generation methods on their percentages of correct assignments, we can also compare the word cloud generation methods from the test person's point of view. For each query, the test persons assess two groups of word clouds without knowing which word cloud generation method was used, and they selected a preference for one of the clouds. The totals of all these pairwise preferences are shown in Table 5. The full-text model performs worst on both tasks. On the subtopic task, the query biased model outperforms the anchor text model, but the difference is not significant.

Analysis To analyze our results and to get some ideas for improving the word clouds we look at the comments of test persons. First thing to be noticed is that test persons pay a lot of attention to the size of the terms in the cloud, and they focus a lot on the bigger words in the cloud. The algorithm we use to determine the font sizes of the terms in the clouds can be improved. Our simple bucketing method works well for log-like probability distributions, but some of the word cloud generation methods like the anchor-text model generate more normal probability distributions. For these distributions, almost all terms will fall into the same bucket, and therefore have the same font size.

One of the most frequently reported problems with the clouds that they contain too much noise, i.e words unrelated to the query. The tolerance of noise differs greatly among the test persons. We can identify three types of noise:

Table 5. Pairwise preferences of test person over word cloud generation models

Model 1	Model 2	# Preferences Subtopic			# Preferences Relevance		
		Model 1	Model 2	Sign test	Model 1	Model 2	Sign test
AN	FT	**47**	21	99%	**43**	23	95%
AN	QB	39	**47**		34	34	
FT	QB	29	**41**		23	**43**	95%

- HTML code. For some queries test persons comment on the occurrence of HTML code in the clouds. This noise can easily be removed by improving the HTML stripping procedure. Since this problem occurs at the document pre-processing step, it affects all word cloud generation methods to the same degree.
- Terms from menus and advertisements. Not all the textual contents of a web page deals with the topic of the web page. Although frequently occurring terms like "Home" or "Search" will be filtered out by our term weighting schemes, sometimes terms from menus or advertisements are included in the clouds. This problem can be solved by applying a content extractor for web pages to extract only the actual topical content of a page such as described in [6]. This procedure can also take care of the HTML stripping. Improving the document pre-processing step will increase the overall quality of all word clouds.
- Non informative terms. Some terms occur frequently in the documents, but do not have any meaning when they are taken out of context, such as numbers (except years). It may be better to not include numbers below 100 and terms consisting of one character at all in word clouds.

This may explain in part why the anchor text clouds work well, that is it has less problems with noise. Anchor text is more focused and cleaner than the full text of a web page.

The second frequently reported problem is that clouds are too similar. During the creation of the user study we already found that clouds created from judged relevant, and judged non relevant documents were very similar. We noticed that the documents judged as non-relevant were very similar in their language use to the relevant documents, so using the judged non-relevant documents led to only minor differences in the language models of the relevant documents and the non-relevant documents. We suspect most search systems that contributed to the pool of documents to be judged are heavily based on the textual contents of the documents, whereas a commercial search engine uses many other factors to decides on the ranking of pages, leading to documents whose textual content will be more dissimilar.

A similar observation is made in the recent work of Venetis et al. [20].They define a formal framework for reasoning about tag clouds, and introduce metrics such as coverage, cohesiveness and relevance to quantify the properties of tag clouds. An 'ideal user satisfaction model' is used to compare tag clouds on the mostly uncorrelated evaluation metrics. A user study is conducted to evaluate the user model. Although the model often predicts the preferred tag cloud when users reach agreement, average user agreement is low. They observe in many cases users do not have a clear preference among clouds, it is therefore important for user studies involving word or tag clouds to make sure there are clear differences between the clouds.

For for some of the queries in our study the clouds are indeed very similar to each other with a large overlap of the terms in the cloud. The query biased clouds emphasise the differences between the clusters of documents, and generate the most dissimilar clouds. This is most probably the reason why the test persons prefer the query biased clouds. Unfortunately, the query bias in the clouds does comes with a loss of overall quality of the clouds and does not lead to a better representation of the topic and the relevance in the clouds.

Summarising the results, anchor text is a good source of information to generate word clouds and although query biased clouds are preferred by the test persons, they do not help to convey the topic and relevance of a group of search results.

5 Conclusions

In this paper we investigated whether word clouds can be used to summarize multiple search results to convey the topic and relevance of these search results. We generate word clouds using a parsimonious language model that incorporates n-gram terms, and experiment with using anchor text as an information source and biasing the clouds towards the query.

The snippets used in modern web search are query biased, and are proven to be better than static document summaries. We want to examine if the same is true for word clouds, hence our first research question is: *Are query biased word clouds to be preferred over static word clouds?* Surprisingly, we have not found any positive effects on the performance of test persons by biasing the word clouds towards the query topic. The test persons however did appreciate this model in their explicit preferences, because it emphasizes the differences between the clusters of documents.

Secondly, we studied the use of anchor text as a document surrogate to answer the question: *Is anchor text a suitable source of information to generate word clouds?* We find a positive answer to this research question; anchor text is indeed a suitable source of information. The clouds generated by the documents' anchor text contain few noisy terms, perform better than the full-text model, and the anchor text clouds are preferred by the test persons as well.

Finally, the main research question of this paper was: *How can we use word clouds to summarize multiple search results to convey the topic and relevance of these search results?*. We have studied a new application of word clouds, and tested how well the user perception of such a cloud reflects the underlying result documents, either in terms of subtopics or in terms of the amount of relevance. Although tag and word clouds are pervasive on the Web, no such study exists in the literature. The outcome of our study is mixed. We achieve moderately positive results on the correspondence between the selected word clouds and the underlying pages. Word clouds to assess the relevance of a complete SERP achieve an accuracy of around 60% of the assignments being correct, while subtopics are matched with an accuracy of around 70%. It is clear however that interpreting word clouds is not so easy. This may be due in part to the unfamiliarity of our test persons with this task, but also due to the need to distinguish between small differences in presence of noise and salient words. Especially the word clouds based on varying degrees of relevant information seem remarkably robust. This can also be regarded as a feature: it allows for detecting even a relatively low fraction of relevant results.

In future work we would like to compare the use of word clouds for summarization of search results to other summarization methods, such as snippets. While for the subtopic matching task a snippet from a single document could be sufficient, for the relevance assessment task we would need to experiment with generating snippets from multiple documents, since the relevance level of a complete result page cannot be judged by a snippet from a single result. We also would like to apply content extraction techniques to extract the actual content from the web pages and thereby reduce the noise occurring in the clouds.

Acknowledgements This research was supported by the Netherlands Organization for Scientific Research (NWO, under project # 612.066.513). In case you are wondering: the correct assignments of the clouds in Figures 1, 3 and 4 respectively are: 1-A, 2-C, 3-B; A-Non Rel., B-Some Rel., C-Rel.; and 1-C, 2-A, 3-B.

REFERENCES

[1] S. Bateman, C. Gutwin, and M. Nacenta. Seeing things in the clouds: the effect of visual features on tag cloud selections. In *HT '08: Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, pages 193–202, New York, NY, USA, 2008. ACM.

[2] D. Carmel, H. Roitman, and N. Zwerdling. Enhancing cluster labeling using wikipedia. In *Proceedings of SIGIR'09*, pages 139–146, New York, NY, USA, 2009. ACM.

[3] C. L. Clarke, N. Craswell, and I. Soboroff. Overview of the trec 2009 web track. In *The Eighteenth Text REtrieval Conference (TREC 2009) Proceedings*, 2010.

[4] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. Domain-specific keyphrase extraction. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 668–673, 1999.

[5] E. Glover, D. M. Pennock, S. Lawrence, and R. Krovetz. Inferring hierarchical descriptions. In *Proceedings of CIKM'02*, pages 507–514, New York, NY, USA, 2002. ACM.

[6] S. Gupta, G. Kaiser, D. Neistadt, and P. Grimm. Dom-based content extraction of html documents. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 207–214, New York, NY, USA, 2003. ACM.

[7] M. J. Halvey and M. T. Keane. An assessment of tag presentation techniques. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1313–1314, New York, NY, USA, 2007. ACM.

[8] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM Press, New York NY, 2004.

[9] R. Kaptein, D. Hiemstra, and J. Kamps. How different are language models and word clouds? In C. Gurrin, Y. He, G. Kazai, U. Kruschwitz, S. Little, T. Roelleke, S. Rüger, and K. van Rijsbergen, editors, *Advances in Information Retrieval: 32nd European Conference on IR Research (ECIR 2010)*, volume 5993 of *LNCS*, pages 556–568. Springer, 2010.

[10] R. Kaptein, P. Serdyukov, J. Kamps, and A. P. de Vries. Entity ranking using Wikipedia as a pivot. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM 2010)*, pages 69–78. ACM Press, New York USA, 2010.

[11] B. Y.-L. Kuo, T. Hentrich, B. M. Good, and M. D. Wilkinson. Tag clouds for summarizing web search results. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1203–1204. ACM, 2007.

[12] P. Pirolli, P. Schank, M. Hearst, and C. Diehl. Scatter/gather browsing communicates the topic structure of a very large text collection. In *Proceedings of the SIGCHI conference on Human factors in computing systems: common ground*, CHI '96, pages 213–220, New York, NY, USA, 1996. ACM.

[13] J. Ponte and W. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st ACM Conference on Research and Development in Information Retrieval*, pages 275–281, 1998.

[14] 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 *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 995–998, New York, NY, USA, 2007. ACM.

[15] M. Song, I. Y. Song, R. B. Allen, and Z. Obradovic. Keyphrase extraction-based query expansion in digital libraries. In *JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 202–209, 2006.

[16] M. Srikanth and R. Srihari. Biterm language models for document retrieval. In *Proceedings of SIGIR'02*, pages 425–426, New York, NY, USA, 2002. ACM.

[17] A. Tombros and M. Sanderson. Advantages of query biased summaries in information retrieval. In *Proceedings of SIGIR'98*, pages 2–10, New York, NY, USA, 1998. ACM.

[18] M. Tsagkias, M. Larson, and M. de Rijke. Term clouds as surrogates for user generated speech. In *Proceedings of SIGIR'08*, pages 773–774, New York, NY, USA, 2008. ACM.

[19] P. Turney. Coherent keyphrase extraction via web mining. In *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 434–442, 2003.

[20] P. Venetis, G. Koutrika, and H. Garcia-Molina. On the selection of tags for tag clouds. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 835–844, New York, NY, USA, 2011. ACM.

[21] R. W. White, I. Ruthven, and J. M. Jose. Finding relevant documents using top ranking sentences: an evaluation of two alternative schemes. In *Proceedings of SIGIR'02*, pages 57–64, New York, NY, USA, 2002. ACM.