Document Word Clouds: Visualising Web Documents as Tag Clouds to Aid Users in Relevance Decisions

Thomas Gottron

Institut für Informatik
Johannes Gutenberg-Universität Mainz
55099 Mainz, Germany
gottron@uni-mainz.de

Abstract. Information Retrieval systems spend a great effort on determining the significant terms in a document. When, instead, a user is looking at a document he cannot benefit from such information. He has to read the text to understand which words are important. In this paper we take a look at the idea of enhancing the perception of web documents with visualisation techniques borrowed from the tag clouds of Web 2.0. Highlighting the important words in a document by using a larger font size allows to get a quick impression of the relevant concepts in a text. As this process does not depend on a user query it can also be used for explorative search. A user study showed, that already simple TF-IDF values used as notion of word importance helped the users to decide quicker, whether or not a document is relevant to a topic.

1 Introduction

Research on Information Retrieval (IR) systems aims at helping users to satisfy their information needs. One important task in the underlying theoretical models is to determine the – generally speaking – important terms in an indexed document. This knowledge is then used to compute the relevance of a document to a given query.

But, once the user has selected a document from the result list of a search engine, he cannot access any longer the IR system's notion of word importance. He has to take a look at the document and judge by himself, whether it really is relevant to his information need. This is even more the case for explorative search, where the user browses along some reference structures and does not formulate a query at all.

Web designers, however, are aware that users usually do not really *read* documents on the web. Initially a user only *scans* a document [1,2]. This means to pass with the eyes over the screen quickly and pick up bits and pieces of layout, images or words here and there. Only if this quick *scanning* provides the user with the impression that it is worthwhile to actually read the entire text, he proceeds.

M. Agosti et al. (Eds.): ECDL 2009, LNCS 5714, pp. 94–105, 2009.

On the other hand, a study on users' information seeking behaviour on the web [3] revealed, that the text contents are the most useful feature in web documents to judge relevance. Unfortunately, when scanning a web document, its plain text parts usually do not attract much attention.

This paper proposes to support the user in the task of visually scanning a document employing techniques similar to the ones used in tag clouds. Based on a TF-IDF model, we calculate the importance of each word in a web document and display it with a respectively larger or smaller font size. These *document word clouds* allow a user to perceive much faster, which words in a document distinguish its content. We implemented a desktop http proxy server to analyse web documents on-the-fly and convert them into word clouds. Hence, users can use this visualisation enhancement transparently while browsing the web. For eventually reading a document, the system allows to convert it back into its normal state. A small experiment showed, that with this kind of document visualisation, users can decide quicker, whether or not a document is relevant to a given topic.

We proceed as follows: after a brief look at related work in 2, we describe our notion of word importance based on TF-IDF in 3. In section 4 we use the word importance to turn web documents into document word clouds. This section also explains the design of our http proxy implementation. In 5 we finally analyse users' experience with this document representation before concluding the paper in 6.

2 Related Work

The study of Tombros, Ruthven and Jose [3] analysed which factors influence a user's relevance decision in a web context. Examining the information seeking behaviour of 24 users, they found out that features in the text and structure of a web document are considered most useful in relevance judgements. Another interesting result: while a document's text in general was mentioned in 44% of the cases to be a useful relevance hint, the sub-categories *titles/headlines* and *query terms* were mentioned only in 4.4% and 3.9% respectively.

Highlighting the query terms in retrieved documents with a coloured background is a common way to support a user in scanning a document for relevance. Google's web search, for instance, is highlighting query terms in documents retrieved from its cache. Ogden et al. [4] analysed how such a highlighting helped users in a relevance judgements in combination with a thumbnail visualisation of the document. Dziadosz and Raman [5] tried to help users to make their relevance decision already a step earlier. They extended a classical result list of a web search engine with thumbnail previews of the documents. The combination of thumbnail and text summaries allowed the users to make more reliable decisions about a document really being relevant to their information need. All these approaches, however, depend on a user formulated query and are not suitable for explorative search.

Tag clouds are a common visualisation method in the Web 2.0 community. They alphabetically list user annotations (tags) of documents, pictures, links or other online contents. The importance of the tags, i.e. how often they have been used to annotate contents, is visualised, too. More important tags are represented with a larger font size or different colours. In this way, tag clouds provide a very quick impression of trends or "hot topics".

Research on tag clouds is relatively scarce. Publications in this context often merely use the tags as a resource for data-mining tasks (e.g. in [6]). The visual effects used in tag clouds were analysed by Bateman, Gutwin and Necenta [7]. They found that font-size, font-weight and intensity had the strongest visual influence on a user's perception of importance. Viégas and Wattenberg [8] discuss tag clouds from the point of view of visualisation techniques. Not originating from a visualisation research background, they say, tag clouds break some "golden rules". However, given their success and wide adoption, one has to recognise their effectiveness.

Our method to determine word importance is based on classical TF-IDF weights. This concept can be found in virtually every standard IR book. For details and further reading we refer to the very good introduction of Manning, Raghavan and Schütze [9]. Though a classical method it is still applied in current research to determine important words in web documents. In [10], document terms are connected to Wikipedia categories to determine the general document topic. The influence of the categories assigned to each term is weighted by TF-IDF values of the terms. So, the authors imply a correlation between the TF-IDF values of the terms and their importance for the topic of the document.

To determine which parts of a document to include in the calculation of word importance, we use content extraction techniques. We adopted the fast TCCB algorithm [11] with optimised parameter settings [12]. For term normalisation we used stemmer implementations provided by the Snowball project (http://snowball.tartarus.org/): the Porter stemmer [13] for English documents and the project's stemmer for German language.

3 Determining Word Importance

Our aim is to highlight those words in a document, which are more important than others, i.e. which distinguish a particular document from other documents.

The concept of word importance in a document can be mapped onto term weighting in an IR system. Effectively, here we are going to use a simple TF-IDF scheme. For each term t we determine its document frequency $df(t)$, i.e. in how many documents of a corpus of size N it appears (we come to the question of which corpus to use in 4.2). For a given document d we then determine the term frequency $tf_d(t)$, i.e. we count how often the term appears in this particular document. The TF-IDF weight for term t in document d is defined as:

$$\text{TF-IDF}_d(t) = tf_d(t) \cdot \log \frac{N}{df(t)}$$

This formula describes a classical weighting scheme for terms in a vector space IR model. If a query term matches an index term with a high TF-IDF value, the corresponding documents obtain a higher relevance score. The intention behind this scoring is, that a term with a high TF-IDF score describes the document very well – especially in comparison to other documents in the corpus. Hence, we adopt TF-IDF values of the terms in a document as a notion of word importance. Note, that we can do this without having the user have formulated a query.

4 Creating Document Word Clouds

The idea of document word clouds is to transfer the visualisation idea of tag clouds to web documents. Instead of visualising tags, we modify the font size of the words contained in the document itself. Instead of using the frequency of tag assignments to determine how large to write a word, we use the above explained notion of word importance. And instead of sorting the terms alphabetically – as done in tag clouds – we leave their order unchanged.

To actually turn a web document into a document word cloud, we implemented an http proxy server for on-the-fly analysis and modification of documents. Embedding the system in a proxy server is a very flexible solution as it is entirely independent of both sides: the browser client and the web server. Figure 1 sketches the system and we proceed with a detailed explanation on how it works.

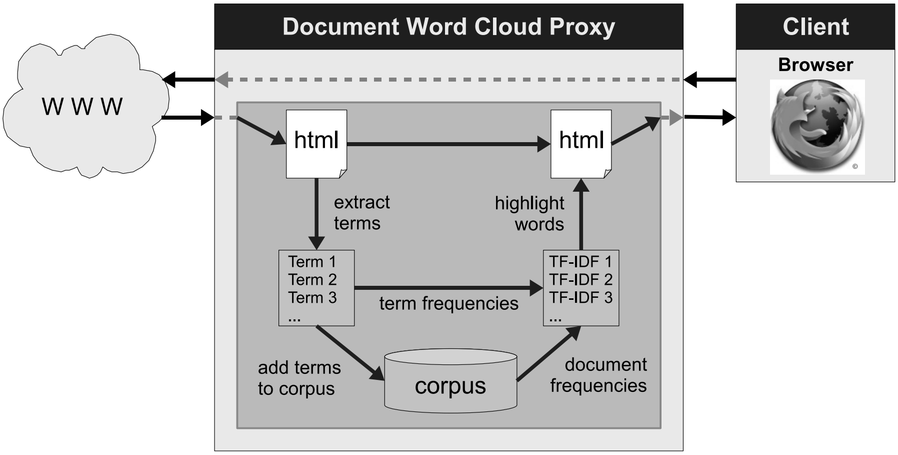


Fig. 1. Our proxy analyses and modifies documents on-the-fly

4.1 Document Preprocessing

After forwarding a client's request to a server and receiving the according response, the proxy does not deliver the document directly back to the client, but first analyses and eventually modifies it.

Therefore, we first need to determine the terms in the document. Exploiting the inherent document structure of HTML, we tokenise the contents into words, using white space characters, sentence and paragraph delimiters and paying attention to some particularities like abbreviations. The resulting tokens are then normalised via case folding. With a simple and fast heuristic we determine the language of the document: we assume the document to be in the language in which it contains the most stopwords. The implemented system so far contains stop word lists for English and German, but can easily be extended to other languages[1]. Once we know which language we are dealing with, we apply a stemmer to finally obtain the terms we use for indexing and TF-IDF calculation.

4.2 Corpus

To calculate TF-IDF for the terms in the document we need a corpus over which to calculate the document frequencies $df(t)$. Actually, as we distinguish between documents in different languages, we need several corpora: one for each language.

One option is to provide a standard corpus for each language. These corpora would have to provide representative frequencies for a large choice of words. This approach has two disadvantages. First, from a practical point of view, it is difficult to provide and handle such a corpus. Probably all available corpora have a bias towards some topics. Further, to store and look up document frequencies in such a large corpus, would result in a higher demand for computational and storage resources. This would make it more difficult to use the system as a desktop proxy and might require a dedicated high-end machine. The second disadvantage is more user specific. Even if we could provide and efficiently handle such an ideal corpus, it might not be suitable for an individual user. If, for instance, a user is mainly interested in a particular topic, the terms in the documents he looks at have very different document frequencies. Say, the user is interested mainly in portable computers. In this case terms like notebook, laptop or netbook are much less informative to the user, than what might be deduced from their document frequencies in a standard corpus.

An alternative is to build the reference corpora while the user browses the web. As the proxy tokenises and analyses each document anyway, we can keep track of the document frequency of each observed term. In this way, we can operate on a corpus that also reflects the user's interests. As in the beginning such a continuously extended corpus is very small considering the number of terms seen so far, the first few browsed documents will be analysed on a basis of pretty distorted document frequencies for the contained words. Accordingly, also the TF-IDF values will be distorted. Hence, a conversion into document word clouds makes sense only after the corpus has reached a certain size. Empirically we found out, that already a corpus of around 3,000 to 4,000 unique terms was sufficient to obtain reasonable results in document visualisation.

Both alternatives have their advantages and disadvantages. For our proxy system we chose the latter option of building the corpus during runtime. However, a third option would be to combine both approaches: to start with a small general corpus and to extend it constantly with newly browsed documents.

[1] Provided the concept of stop words exists in these languages.

4.3 Document Rewriting

On the technical side of actually rewriting a document *d* we first need to calculate the actual TF-IDF values of its terms. The term frequencies $tf_d(t)$ are obtained by counting the term appearances in the document. The document frequencies $df(t)$ are stored in central data structure along with the corpus' size. This is all the data needed to compute the importance of a particular word in the document.

Once we know the importance of each word in the document, we can turn the document into a word cloud. Those words with a relative high importance, i.e. TF-IDF value, will be written larger, while those with low values are written smaller.

To obtain always similar results in font size and in the distribution of large and small words, we normalise the TF-IDF values into *k* classes. Within each of these importance classes, we will display all words with the same font size. The terms with the lowest importance are always assigned to the lowest class, the highest TF-IDF value in the document corresponds to the highest class. The parameter *k* can be set by the user. For our user test we found a setting of $k = 10$ to provide enough importance classes.

The assignment of terms into the classes follows a logarithmic scheme. Given the highest TF-IDF value $w_{\max}$ and the lowest value $w_{\min}$, a term with a TF-IDF value of *t* is assigned to the class:

$$class(t) = \lfloor \left(\frac{t - w_{\min}}{w_{\max} - w_{\min}} \right)^{\beta} \cdot k \rfloor$$

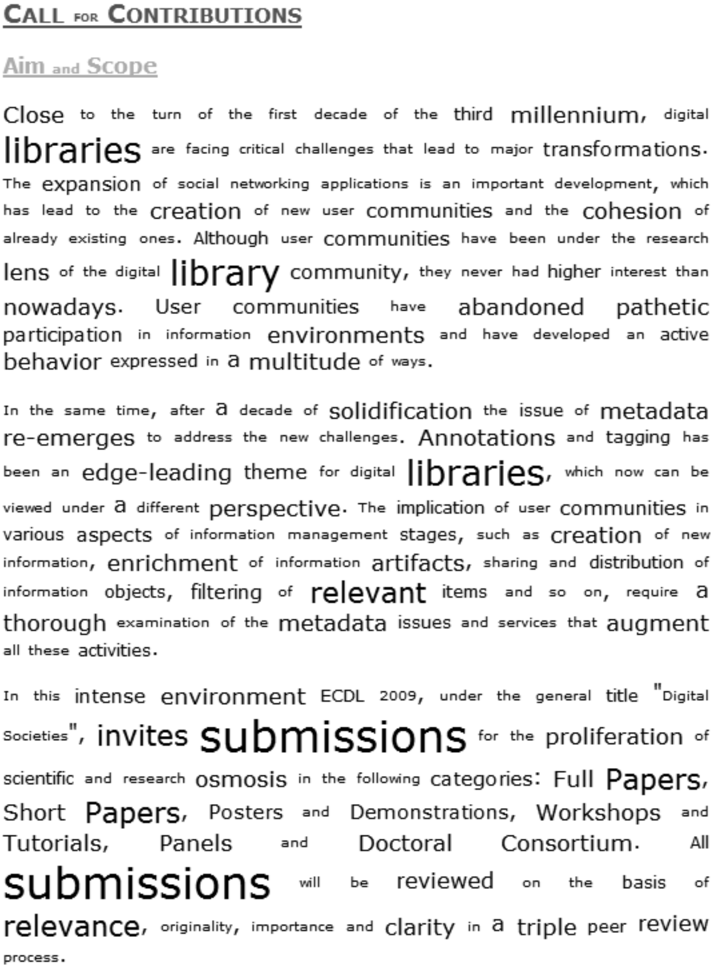
The parameter β influences the distribution into the importance classes. The higher the value the smaller is the proportion of larger written words. In our tests we used a value of $\beta = 1.2$ which produced well balanced document word clouds.

In order to change the font size, each word in the document body is wrapped into `span` tags. These `span` elements all have a `class` attribute with a value representing the importance class of the contained word. Once the words are marked in this way, the document is extended with CSS directives. The CSS causes the font size of the texts inside the `span` elements to be increased or decreased according to the importance class and relative to its normal font size. After these transformations the proxy serialises the document into an http message and returns it to the client who requested it in the first place.

Concerning runtime, the whole process is unproblematic. The analysis of the document, its tokenisation into terms, computation of TF-IDF values, the annotation of the words, the document's extension with CSS directives and its serialisation usually takes less than a second. So, the users do not feel big delays when browsing the web via our proxy.

4.4 The User's Side

In the client browser the documents look like the one shown in figure 2. The screenshot demonstrates nicely how important words are written larger, while, for instance, stop words are very small.



CALL FOR CONTRIBUTIONS

Aim and Scope

Close to the turn of the first decade of the third millennium, digital libraries are facing critical challenges that lead to major transformations. The expansion of social networking applications is an important development, which has lead to the creation of new user communities and the cohesion of already existing ones. Although user communities have been under the research lens of the digital library community, they never had higher interest than nowadays. User communities have abandoned pathetic participation in information environments and have developed an active behavior expressed in a multitude of ways.

In the same time, after a decade of solidification the issue of metadata re-emerges to address the new challenges. Annotations and tagging has been an edge-leading theme for digital libraries, which now can be viewed under a different perspective. The implication of user communities in various aspects of information management stages, such as creation of new information, enrichment of information artifacts, sharing and distribution of information objects, filtering of relevant items and so on, require a thorough examination of the metadata issues and services that augment all these activities.

In this intense environment ECDL 2009, under the general title "Digital Societies", invites submissions for the proliferation of scientific and research osmosis in the following categories: Full Papers, Short Papers, Posters and Demonstrations, Workshops and Tutorials, Panels and Doctoral Consortium. All submissions will be reviewed on the basis of relevance, originality, importance and clarity in a triple peer review process.

Fig. 2. Part of the ECDL 2009 call for contribution converted into a document word cloud

For the purpose of reading the document, the proxy additionally embeds some JavaScript and HTML code. A small button calls a JavaScript function which resets the font sizes of all words to their normal value. This *zooming back* is realised in a smooth, gradual way. Such a smooth transition to the normal document representation turned out to be less confusing for the user.

Alternatively to the embedded button, the proxy can also be configured to let the documents zoom back into their normal state automatically after a preset timeout.

4.5 Optimisation

When analysing our first document word clouds we noticed some unwanted effects. Particularly those terms in headlines showed relatively high document frequencies. Hence, they were displayed extraordinarily small. The reason was, that headlines are often used as anchor texts in the navigation menus or related links lists of other documents. Similarly, the terms in navigation menus also distorted the term frequencies within a document and caused insignificant words to be displayed larger. In other cases, this happened because a term appeared in a legal disclaimer or in advertisements.

To overcome these problems, we restricted the analysis of term frequency and document frequency to those parts of the documents which contain the main text content. Therefore, we used the TCCB content extraction algorithm [11,12]. Though, TCCB does not always outline the main content precisely, its inclusion in the preprocessing phase helped to overcome the problems we observed. Further, we also used it to determine whether a document contains a long text content at all. If the main content is composed of too few words, we do not modify the document at all, as it does not have a textual main content or the text is too short for a reasonable analysis.

5 User Experience

In order to find out, whether the document word cloud approach really supports users in relevance decisions, we conducted a small user test. The users were shown a series of documents in a normal or a word cloud representation. Their task was to decide as fast as possible if a document belonged to a given topic. So, the users had to make a simple yes/no decision. They were not told, that larger words in the document word clouds represented significant words. In order to find out, how fast and reliable the users could judge the relevance of a document for a given topic we measured the time they needed to make their decision and whether their decision was correct.

As documents we used articles taken from a German online news website. The corpus for calculating word importance was a larger collection of 1,000 news articles from the same website. All documents consisted of a headline, a teaser paragraph and the full article body. The topic categories were cars (test reports), cell phones (hardware, service providers), economics (international economic policies, large companies) and tv/movies (actors, directors, films). For all documents it was rather clear whether they actually belonged to the topic or not. However, for some documents already the headline contained good hints for a decision, for others it was necessary to read parts of the article body.

We had 14 participants in the test. They were all familiar with documents in a web environment, several knew the concept of tag clouds. Each user was given five documents for each of the four topics. So, they had to judge a total of 20 documents each. We divided the users in two groups of equal size. The first group was provided with documents in the standard format (large headline, a bold written teaser paragraph and plain text for the article) for the topics tv/movies and economics, while the document for the topics cars and cell phones were shown as document word clouds. Group two had the same documents, but based on the respectively other presentation style.

Table 1. Time needed for relevance decisions (in seconds). Group 1 saw the categories cars and cell phones as document word clouds, group 2 the categories tv/movies and economics.

		user group 1							user group 2						
	documents	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14
tv/movies	D1	3.01	3.30	2.39	2.18	2.96	2.54	3.93	2.48	1.68	1.65	2.99	1.66	2.96	1.32
	D2	2.46	2.25	2.35	2.20	1.87	2.67	2.06	7.77	3.58	1.51	2.30	2.36	5.88	2.29
	D3	2.61	3.48	1.69	3.69	2.11	4.86	3.25	3.97	3.86	1.43	3.19	2.68	2.97	2.62
	D4	2.22	3.94	3.13	6.67	4.26	4.16	4.77	3.26	2.75	1.71	2.13	3.77	3.20	1.81
	D5	3.14	8.17	3.78	2.13	2.41	2.23	3.34	3.39	2.20	1.75	3.28	4.25	2.63	4.43
economics	D6	1.99	2.48	3.00	4.27	1.68	2.54	2.21	2.68	2.96	1.72	5.20	3.49	2.46	2.72
	D7	2.51	2.99	2.47	2.32	1.61	2.63	1.73	2.64	1.78	1.29	2.30	2.06	1.42	1.42
	D8	3.27	3.65	2.60	2.55	1.42	3.86	2.42	3.55	4.01	1.57	2.46	2.36	2.58	3.81
	D9	2.59	3.79	2.57	2.91	2.23	6.79	3.11	2.82	6.44	1.38	2.50	3.62	1.47	4.51
	D10	2.64	2.50	3.85	3.21	1.92	4.20	3.45	4.44	3.66	1.78	3.57	3.15	2.06	3.20
	avg.	*2.64*	*3.65*	*2.78*	*3.21*	*2.25*	*3.65*	*3.03*	*3.70*	*3.29*	*1.58*	*2.99*	*2.94*	*2.76*	*2.81*
cars	D11	4.42	3.11	3.64	2.96	1.21	2.40	2.89	4.36	3.53	2.77	3.88	2.80	3.77	3.76
	D12	3.35	5.04	2.82	2.71	1.55	2.42	3.37	2.64	2.59	2.01	3.79	3.55	3.43	2.68
	D13	2.92	3.90	1.60	2.43	1.47	4.03	3.69	3.95	3.01	2.75	4.24	5.52	3.91	2.68
	D14	1.76	1.84	1.48	1.78	1.53	2.90	2.48	2.97	3.11	2.19	2.06	2.49	2.10	3.52
	D15	2.78	2.18	2.37	1.19	1.35	3.14	2.19	2.36	3.65	1.67	2.44	2.02	2.44	2.13
cell phones	D16	3.47	5.96	2.68	2.47	2.62	3.36	4.94	8.70	6.38	2.22	2.93	11.11	3.10	3.06
	D17	2.78	2.48	1.76	1.42	1.63	1.76	1.88	1.84	2.23	1.25	1.73	2.15	2.07	1.73
	D18	2.19	2.03	1.42	2.02	1.98	2.32	2.34	2.17	4.01	1.64	2.04	3.56	1.85	2.17
	D19	2.17	2.58	1.92	2.53	2.15	2.51	3.34	2.15	2.12	1.36	4.44	2.78	2.67	3.82
	D20	2.50	3.13	2.50	1.85	1.53	4.28	2.64	3.28	2.00	1.51	3.35	3.37	3.67	2.19
	avg.	*2.83*	*3.23*	*2.22*	*2.14*	*1.70*	*2.91*	*2.98*	*3.44*	*3.26*	*1.94*	*3.09*	*3.93*	*2.90*	*2.77*
	tendency	-	+	+	+	+	+	0	-	0	+	0	+	+	0

Table 1 lists the time the users needed for their relevance decision. It lists the users as U1 to U14 and the documents D1 to D20 for the purpose of reference in the following discussion of the results. The table also indicates the tendency, whether a user made his decision faster (+), slower (-) or more or less in the same time (0) when presented with document word clouds. Note, that the second user group actually saw the documents D11 to D20 first and in their normal representation, before getting documents D1 to D10 as word clouds.

We can observe, that document word clouds allow a quicker relevance decision. On a global average, the candidates took their relevance decision on document word clouds about 0.32 seconds faster. Given the average decision time of 3.1 seconds on the normal document format, this means a 10% quicker decision. Looking at the average time each individual user needed to make her or his judgement in figure 3, we see that two users (U1, U8) took longer for their decision with the cloud representation, four (U7, U9, U11, U14) took more or less the same time and eight were faster. For five users the improvements are statistically significant.

Also with the focus on the single documents in figure 4 the improvement can be measured. For two documents (D2, D5) the decision took longer if they were

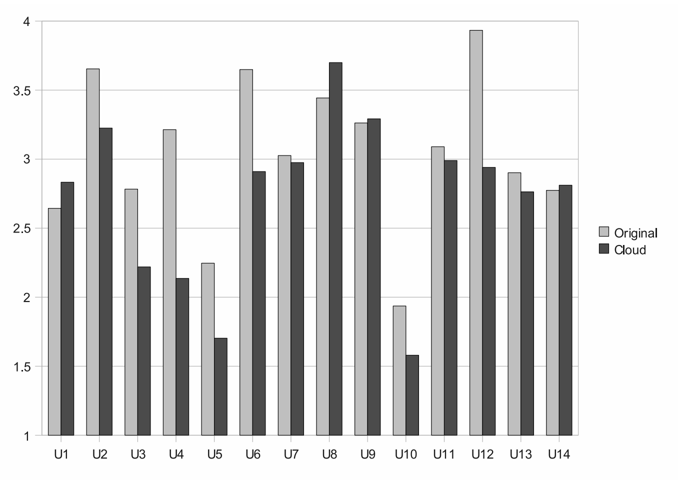


Fig. 3. Average time users needed for relevance judgements on original documents and document word clouds

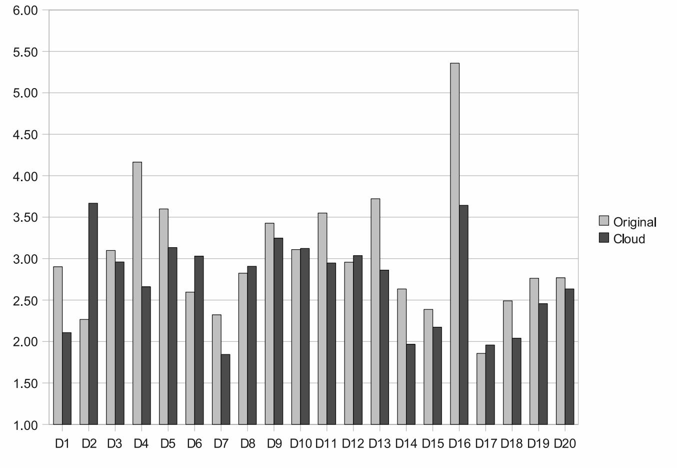


Fig. 4. Average time needed for the documents to be judged depending on their representation in original or cloud format

presented as document word clouds, for four (D8, D10, D12, D17) there was no notable difference, and for the remaining 14 the responses came quicker. Document D2 took significantly longer to assess relevance in its cloud representation. The problem was, that most of the highlighted words (though specific for the actual content of the document) did not allow a clear negative answer, which was expected and eventually given for this document. However, for five documents the time improvements are statistically significant.

In following-up interviews, most users said, the larger font-size attracted their attention and made them focus on theses words. Accordingly, they also mentioned the importance of highlighting those words in a document, which actually aid a decision. Some users (U1, U3, U8, U9) said, they did not “trust” the larger words. So, they tried to read the text in the standard fashion left to right and top down. As we did not provide the option to switch to the normal representation during the test, reading was more difficult and most of those users took longer or were at least not faster in their decision.

The users made relatively few mistakes in their decision. Only two misclassification were made on the original documents, and four on the cloud representation. Three of the latter mistakes occurred in the topic of economy news, because an off-topic document (D9) about operating systems mentioned Apple and Microsoft. In this case, the users took those names as a hint for a business report about those companies. Also other users mentioned, that names of companies or persons are not always useful in a relevance judgement. So, their usually high TF-IDF values do not necessarily correspond to their importance for relevance assessment.

Finally, some users said, they were initially confused by the unusual cloud layout of the document. However, they also said, that given a little more time they would probably get used to it – and then it might become very helpful.

6 Conclusions

The concept of document word clouds helps users to get a quick idea of what a document is about. Inspired by tag clouds we write important words using a larger font-size, while reducing the font-size of less significant words. As a measure for word importance we used TF-IDF values. The whole process is independent of a user query, hence, can also be used for explorative search. We implemented a http proxy to convert web documents on-the-fly into document word clouds. The system was used to create documents for a user test, in which document word clouds allowed the users to make relevance decisions quicker.

The developed system can be extended and improved in several directions. First of all, it would be interesting to realise document word clouds on different, more sophisticated IR models. Techniques to detect named entities can be used to recognise names of persons or companies. They usually have a high TF-IDF values but are not always supportive for relevance decisions. Another problem are longer documents in which the important words appear towards the end. In this case, the users do not see any large written words unless they scroll down. A summary at the top or an embedded thumbnail might solve this problem. A time-limit for keeping documents in the corpus might be an idea for future research, as well. It could improve the systems performance when the interest of a user changes. More practical issues would be to finetune the determination of suitable relevance classes and to include a duplicate or near-duplicate detection to avoid indexing the same document more than once.

References

1. Krug, S.: Don't make me think – Web Usability, 2nd edn. mitp, Heidelberg (2006)
2. Lindgaard, G., Fernandes, G., Dudek, C., Browñ, J.: Attention web designers: You have 50 milliseconds to make a good first impression! Behaviour & Information Technology 25(2), 115–126 (2005)
3. Tombros, A., Ruthven, I., Jose, J.M.: How users assess web pages for information seeking. J. Am. Soc. Inf. Sci. Technol. 56(4), 327–344 (2005)
4. Ogden, W.C., Davis, M.W., Rice, S.: Document thumbnail visualization for rapid relevance judgments: When do they pay off? In: TREC, pp. 528–534 (1998)
5. Dziadosz, S., Chandrasekar, R.: Do thumbnail previews help users make better relevance decisions about web search results? In: SIGIR 2002: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 365–366. ACM, New York (2002)
6. Noll, M.G., Meinel, C.: Exploring social annotations for web document classification. In: SAC 2008: Proceedings of the 2008 ACM symposium on Applied computing, pp. 2315–2320. ACM, New York (2008)
7. Bateman, S., Gutwin, C., Nacenta, M.: Seeing things in the clouds: the effect of visual features on tag cloud selections. In: HT 2008: Proceedings of the nineteenth ACM conference on Hypertext and hypermedia, pp. 193–202. ACM, New York (2008)
8. Viégas, F.B., Wattenberg, M.: Tag clouds and the case for vernacular visualization. Interactions 15(4), 49–52 (2008)
9. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
10. Schönhofen, P.: Identifying document topics using the wikipedia category network. In: WI 2006: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 456–462. IEEE Computer Society Press, Los Alamitos (2006)
11. Gottron, T.: Content code blurring: A new approach to content extraction. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2008. LNCS, vol. 5181, pp. 29–33. Springer, Heidelberg (2008)
12. Gottron, T.: An evolutionary approach to automatically optimise web content extraction. In: IIS 2009: Proceedings of the 17th International Conference Intelligent Information Systems (in preparation, 2009)
13. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)