Generating snippets for undirected information search
Arnold Overwijk
arnold.overwijk@gmail.com

ABSTRACT
In this paper a system is described to generate snippets for undirected information search. The system is divided in three components: extracting snippets, ranking snippets and avoiding redundancy. To determine the best methods for these components, different experiments have been carried out using datasets from the Cross Language Evaluation Forum. Unfortunately evaluation turned out to be impossible due to a wrong evaluation method and dataset of WebCLEF 2007. This paper shows the problems of the evaluation used in WebCLEF 2007 and discusses possible solutions.

Keywords
Single document summarization, snippet generation, query biased summaries, undirected information search

1. INTRODUCTION
Nowadays the amount of information on the net is enormous and this amount is ever growing. This makes search engines a vital tool for finding information. Since almost every query gives more results than the user would be able to read, it is essential that the most relevant documents are shown first. But most of the time these documents contain only a small amount of relevant information for the user. This means the user often has to take a look at the whole document for only a single snippet of information, which is very time consuming. Therefore search engines generate snippets (i.e. the query terms that appear in the document and the words around them) to give the user a ‘sneak preview’ of the document’s content [1]. Snippets help the user to assess the relevance of the document without accessing it. The generation of snippets is mainly based on the query terms and less on the context of the document (i.e. fragments in the document that do not contain query terms). Unfortunately this is only sufficient for making relevance decisions [2], which is the intention of search engines. However when the user is unfamiliar with the topic or seeks background information, context has shown to be more important [3]. This means different kind of snippets are useful for different goals. An overview of the different search goals of the users of search engines can be found in [4].

The main reason for users to search is to ‘find out about’ their search topic. Therefore snippets should be more based on the context of the document. Such a goal is known as undirected information search. With over 23% of all queries, undirected information search goals are most common [4]. This paper will be about generating snippets for undirected information search. In the best case this would obviate the need to open any document, because the snippet answers to the information need of the user.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission.

9th Twente Student Conference on IT, Enschede, June 23th, 2008
Copyright 2008, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science

Users that are not familiar with the topic want to make sense of each page to help them reformulate their search and better understand their topic [2]. This means they don’t know what they are looking for and every bit of new information will be useful, which results in many ways that may satisfy the user. Experts on the other hand know what they are looking for and therefore have a specific need for information. Therefore there will only be one feasible solution consisting of the real information need.

Generating snippets means extracting snippets from the documents and weigh these snippets according to their relevance. A snippet is relevant when it contains one or more distinct atomic facts that the user would use for writing an article. Generating these snippets requires a method to extract snippets, a ranking mechanism and similarity measurement to avoid redundancy. Which raises the following questions:

- How to extract a snippet?
- How to rank a snippet?
- What is a suitable method for measuring similarity?

The proposed approach is to compare different methods to find an answer to these questions.

In this paper first an overview of the state of the art on the field of automatic summarization is given. Then an experimental setup is described. This would be evaluated using the evaluation method and dataset of WebCLEF 2007 to find an answer to the research questions. However during the evaluation it turned out that the used evaluation method and dataset is inappropriate. This will be discussed and some possible solutions are presented.

2. RELATED WORK
Automatic document summarization can be divided in single- and multi-document summarization. There are two major differences between single- and multi-document summarization. First, sentence extraction is a particular approach for single-document summarization. It is somewhat difficult to use this approach for multi-document summarization since information is stored in multiple documents possibly with overlap. Second, most single-document summarization systems make use of the monolithic structure of the document. For multi-document summarization the structure of a single document cannot be readily used, therefore these systems usually rely less on the structure of documents [5].

Document summaries can be abstracts or extracts. An extract summary consists of sentences extracted from a document, while an abstract summary may contain words and phrases which do not exist in the original document [6]. Snippet generation is a special type of extractive document summarization [1].

Research in automatic text summarization can be separated in generic and user-directed summaries [7]. Generic summaries focus on the essence of the document and ignore the user, whereas user-directed summaries focus on the information need as expressed by the user. Techniques for generic summaries try to determine the most important sentences from a document based on indicators in the document itself. The analysis takes
generating only one snippet per document is that a document may contain more than one aspect. If this is the case then only sentences that are diverse as possible to prevent redundancy, but still related to the query.

3. EXPERIMENTAL SETUP

Generating snippets for undirected information search requires three steps: extract snippets, rank them and avoid redundancy. The combination of these three steps make a system that can generate snippets given a document collection. The output of this system will be evaluated using the WebCLEF data of 2007.

3.1 System

3.1.1 Extracting snippets

Extracting snippets requires choosing a building block, which can be a word, sentence or even a whole paragraph and in addition a snippet size that is effective for undirected information search. One or more building blocks, depending on the preferred size, form a snippet. Note that these building blocks must be from the same document, since an opportunity to read the whole document should be given to the user.

Using sentences as a building block is the most usual approach in automatic summarization. Unfortunately a sentence often does not contain enough information to be a snippet on its own. Therefore a snippet should be built of multiple sentences, which can be done by summarizing a single document. A paragraph on the other hand does provide enough information. For this reason another slightly different variant of the first ranking mechanism will be tested, that differs from the previous one in similarity measurement, which is not based on known sources, but on the search results. All these ranking mechanisms are combinations between user directed and generic mechanisms.

3.1.2 Ranking snippets

After snippet extraction the next step will be ranking all these snippets according to their relevance for the user. When the collection of candidate snippets is treated as a document collection then the similarity with the search query is the most obvious method to use. The ranking mechanism of last year’s best performing system at WebCLEF [14] is based on similarity with known sources, provided by the user as a part of the search query:

$$\text{let } c_1, \ldots, c_n \text{ be candidate snippets}$$

$$\text{let } k_1, \ldots, k_m \text{ be known snippets}$$

$$\text{for each candidate snippet } c'$$

$$\text{let } \text{score}(c) = \text{score}(c) + \text{sim}(c, c')$$

$$\text{for each known snippet } k$$

$$\text{let } \text{score}(c) = \text{score}(c) - \text{sim}(c, k)$$

$$\text{for each known snippet } k$$

$$\text{if } \text{sim}(c, k) > \text{sim}_{\text{MAX}}$$

$$\text{let } \text{score}(c) = 0$$

The ranking mechanism above is based on the documents which are relevant according to the user, which makes it more user directed. The mentioned ranking mechanism will be compared with a slightly different ranking mechanism that also takes the topic description into account, which makes it even more user directed:

$$\text{let } d \text{ be the free text description}$$

$$\text{let } c_1, \ldots, c_n \text{ be candidate snippets}$$

$$\text{let } k_1, \ldots, k_m \text{ be known snippets}$$

$$\text{for each candidate snippet } c$$

$$\text{let } \text{score}(c) = \text{sim}(c,d)$$

$$\text{for each candidate snippet } c'$$

$$\text{let } \text{score}(c) = \text{score}(c) + \text{sim}(c,c')$$

$$\text{for each known snippet } k$$

$$\text{let } \text{score}(c) = \text{score}(c) - \text{sim}(c,k)$$

$$\text{for each known snippet } k$$

$$\text{if } \text{sim}(c,k) > \text{sim}_{\text{MAX}}$$

$$\text{let } \text{score}(c) = 0$$

Both ranking mechanisms make use of a list with known sources and/or a topic description, which is not available most of the times (e.g. a search in Google). Therefore it is interesting to investigate if the above ranking mechanisms perform better than a ranking mechanism that does not make use of this information. For this reason another slightly different variant of the first ranking mechanism will be tested, that differs from the previous one in similarity measurement, which is not based on known sources, but on the search results. All these ranking mechanisms are combinations between user directed and generic mechanisms.

3.1.3 Measuring similarity

Similarity measurement is by far the most important feature in this system, since it is used in the ranking mechanism as well to avoid redundancy. Therefore different methods will be compared. The most commonly used methods are word overlap using the standard Jaccard coefficient and vector space
similarity or a variant [18]. The Jaccard coefficient, TF.IDF and IDF-only will be compared with each other.

3.1.3.1 Jaccard coefficient

The Jaccard coefficient computes word overlap between two snippets using the standard Jaccard coefficient:

\[
SIM_{\text{Jaccard}}(x, y) = \frac{|x \cap y|}{|x \cup y|}
\]

Where \(x\) and \(y\) are sets of non-stopwords of snippets \(x\) and \(y\) respectively.

The main advantage of this method is that it is extremely simple and therefore very efficient. A disadvantage is that it does not take word frequency into account, which is a problem. For example:

\[
SIM_{\text{Jaccard}}(\{a a a a a b', b'\}) = \frac{1}{\frac{1}{2}} = \frac{1}{2}
\]

Here ‘a’ and ‘b’ are non-stopwords. This problem probably occurs when the snippets are large and about the same topic, because then both snippets may contain important words at least once, but one of the snippets can be about aspect \(a\) while short mentioning aspect \(b\) and the other about aspect \(b\) only.

3.1.3.2 TF.IDF

TF.IDF is a term weighting scheme that includes term frequencies and inverse document frequencies, which is implemented as the cosine of the angle between the vector representations of the snippets computed using the standard TF.IDF weighting scheme [19]:

\[
SIM_{\text{TF.IDF}}(x, y) = \frac{\sum_{\text{each word}} \text{frequency}(x) \times \text{frequency}(y)}{\sqrt{\sum_{\text{each word}} \text{frequency}(x)^2} \times \sqrt{\sum_{\text{each word}} \text{frequency}(y)^2}}
\]

The main advantage of this method is that word frequency is taken into account and the similarity is normalized over the length of the document (i.e. the snippet). The disadvantage of this method in this particular situation is that snippets are relatively small and therefore most words occur only once or twice. This means that the few words that accidently do occur multiple times are over weighted.

3.1.3.3 IDF-only

IDF-only is similar to TF.IDF, except that here only IDF is used instead of TF.IDF. Only using IDF avoids overweighting words that accidently occur multiple times.

3.2 Evaluation method

The Cross Language Evaluation Forum has a Multilingual Web Track called WebCLEF. In WebCLEF undirected informational search goals in a web setting are considered. The purpose in this track is supporting a user who is an expert in writing a survey article on a specific topic with a clear goal and audience. The support will consist of a ranked list with relevant snippets. The degree to which the information need is satisfied is measured by the user as number of distinct atomic facts that the user includes in the article after analyzing top snippets returned by the system. Such a method will be perfect for this research, since the goal of the user here is undirected information search.

3.2.1 Data

WebCLEF provides the following information about the needs of the user:

- A list of known sources: online resources that the user considers to be relevant to the topic and from which may information already have been included in the article.
- Optional list of Google retrieval queries that can be used to locate the relevant information; the queries may use site restrictions to express user’s preferences.

For every retrieval query also the top 1000 hits from Google is available. WebCLEF also provides some information about these documents:

- The rank in the Google result list.
- The Google snippet.

There is also a manual assessment available for each topic which will be used to evaluate the response of the system. The response of the system contains a ranked list per topic with generated snippets.

3.2.2 Evaluation measures

To evaluate the performance of the system, the first 7000 bytes of the output of the system (i.e. a ranked list of snippets) will be compared with the manual assessments, resulting in recall and precision [20].

- **Recall** as the sum of character lengths of all spans in the response of the system linked to nuggets (i.e. an aspect the user includes in his article), divided by the total sum of span lengths in the responses for a topic in all submitted runs.
- **Precision** as the number of characters that belong to at least one span linked to a nugget, divided by the total character length of the system’s response.

These two measures reflect the quality of the system in the following way:

- More redundancy in the output results in a lower precision, since the length of the system’s response increases while the overlap with the manual assessments is the same.
- The better the performance of the similarity method that is used to compare snippets the higher the recall will be, because a better similarity method results in a more accurate snippet score.

The choice of the building block is reflected by recall and precision. Larger units give a higher recall and lower precision, because they have more characters and therefore a higher probability to cover more aspects together with more non-related information.

4. EVALUATION

The data and evaluation method of WebCLEF 2007 is used to analyze different combinations of extracting snippets, ranking mechanisms and similarity methods. During this evaluation only the first 50 documents, given by Google, per query are taken into account. The reason for this is that the implemented ranking mechanism has a complexity of \(O(n^2)\) (where \(n\) is the number of candidate snippets), which is very large when all documents are taken into account. It is likely that the results will be even better than when all documents are used, since the search query is processed by Google and the most relevant documents are taken into account.

The system consists of three parts which all have three options. This leads to \(3^3\) combinations, which is quite a lot because every combination needs a day to run. Therefore a selection is made starting with a system that is most equal to the best
performing system of WebCLEF 2007. This system is used as a baseline. All other systems differ exactly one option, so the impact of this option can be measured. The different evaluated combinations are given in the table below. Here the UvA system is the original system, provided by CLEF, of the UvA (i.e. the best performing system) at WebCLEF 2007.

<table>
<thead>
<tr>
<th>Table 1. System components</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Snippet extraction</strong></td>
</tr>
<tr>
<td>Paragrapheks (P)</td>
</tr>
<tr>
<td>Sentences (S)</td>
</tr>
<tr>
<td>SUMavg (A)</td>
</tr>
<tr>
<td><strong>Ranking mechanism</strong></td>
</tr>
<tr>
<td>UvA (U)</td>
</tr>
<tr>
<td>UvA + description (D)</td>
</tr>
<tr>
<td>UvA based on candidate snippets (C)</td>
</tr>
<tr>
<td><strong>Similarity method</strong></td>
</tr>
<tr>
<td>Jaccard (J)</td>
</tr>
<tr>
<td>TF.IDF (T)</td>
</tr>
<tr>
<td>IDF-only (I)</td>
</tr>
</tbody>
</table>

The results of this evaluation can be found in the table below.

<table>
<thead>
<tr>
<th>Table 2. Evaluation results</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System</strong></td>
</tr>
<tr>
<td>UvA</td>
</tr>
<tr>
<td>PUI</td>
</tr>
<tr>
<td>PUT</td>
</tr>
<tr>
<td>PUI</td>
</tr>
<tr>
<td>PDI</td>
</tr>
<tr>
<td>PCI</td>
</tr>
<tr>
<td>SUI</td>
</tr>
<tr>
<td>AUI</td>
</tr>
</tbody>
</table>

According to these results it is clear that the PUI system differs from the UvA system, since the performance is only half as good as the UvA system. Analysis shows that the system differs at two points, namely the tokenizer and the filtering of stopwords. In the UvA system the implementation of filtering the stopwords contains a bug which causes that only half of the stopwords is filtered at random. It is clear that this is an unwanted bug. To show this, the system without the stopword bug will be evaluated, which should give a better result. To be sure that filtering stopwords is a good approach, also another variant that does not filters any stopwords at all is evaluated. The results of this evaluation is shown in table 3.

<table>
<thead>
<tr>
<th>Table 3. Evaluation results</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System</strong></td>
</tr>
<tr>
<td>UvA</td>
</tr>
<tr>
<td>UvA without bug</td>
</tr>
<tr>
<td>UvA without filtering stopwords</td>
</tr>
</tbody>
</table>

It is obvious that the UvA system without the bug in the implementation of filtering stopwords gives a lower precision and recall. However the precision and recall values are now close to the PUI system. Based on these two observations the expectation is that using no stopwords at all should perform better, which is definitely not the case. Therefore it sounds reasonable that something is wrong with the evaluation of WebCLEF 2007. There are two possible causes, namely the dataset and the method of evaluation. To investigate this, the output and results of the UvA system as well the manual assessments provided by CLEF have to be analyzed.

Analyzing the output and the results of the UvA system indicated that for some topics the output is simply the candidate snippets ordered by occurrence (i.e. the first snippet is the first paragraph of the first document, the second snippet is the second paragraph of the first document, etc.). To measure this a system that orders the snippets according to their occurrence is evaluated. A selection of the results of this experiment is shown in table 4.

<table>
<thead>
<tr>
<th>Table 4. UvA compared with first occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UvA</strong></td>
</tr>
<tr>
<td><strong>Precision</strong></td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>25</td>
</tr>
<tr>
<td>26</td>
</tr>
<tr>
<td>Avg.</td>
</tr>
</tbody>
</table>

In the table above the precision and recall values of both systems are exactly the same for the selected topics. This is because the search query that belongs to these topics did not provide known sources and therefore the UvA system was not able to calculate IDF values, which results in a snippet score of zero for all snippets. These results indicate that the manual assessments are not trustworthy. This is illustrated by the following example, which is actually topic 14 from the manual assessments file:
Here the user provided the following topic description:

Are there any blog search engines in Europe that are not subsidiaries of big three search engines (Google, Yahoo!, Microsoft)? Find home pages of these search engines, and, if possible, short descriptions of their mission. Lists or URLs of blogging platforms are not an appropriate response, unless they include search facilities for more than just their own platform.

We can see that in the example above the manual assessments contain only three different spans and some of them are not even relevant for this topic. There are also topics for which the manual assessments do not even contain snippets, which automatically results in a precision and recall of zero. Another fault is that some snippets are marked as known while they are not. In addition special characters in the search queries and documents disappear, which is especially a problem in languages like Spanish.

After analyzing the evaluation method it turned out that a snippet in the manual assessments should be a part of the output of the system. If this is not the case then there is no match at all, even when there is only one word lacking. To illustrate this problem with the evaluation method, the UvA system is modified in a way that it does not output the last word of a snippet. The result of this experiment is shown in table 5.

<table>
<thead>
<tr>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>UvA</td>
<td>0.2018</td>
<td>0.2561</td>
</tr>
<tr>
<td>UvA last word removed</td>
<td>0.0597</td>
<td>0.0758</td>
</tr>
</tbody>
</table>

From these results we can conclude that a system which provides large snippets has a benefit when this evaluation method is used. Which is definitely not a desired effect.

Unfortunately the mentioned problems make automatic evaluation of different methods for extracting snippets, ranking snippets and measuring similarity impossible, therefore a manual view on the results is done. This shows that for some topics the UvA system returns mess while a system with a ranking mechanism based on candidate snippets provides related snippets. For the other topics it is hard to say which method performs better since they all provide related snippets. This indicates that a system with a ranking mechanism that is more based on candidate snippets performs better, at least for some topics. An logical explanation for this observation can be found in the fact that not all topics provide known sources, but the mentioned observation is even the case for some topics that do provide known sources.

5. DISCUSSION

Analysis of the system results and evaluation of WebCLEF 2007 shows that the dataset as well as the evaluation method of WebCLEF is inappropriate and therefore not suitable to use for evaluation. The problems with the dataset are probably the cause of the fact that there were only three participating systems at WebCLEF 2007. This means the manual assessments are for one third based upon the output of the UvA system. The consequence is that the dataset is not usable, since a system that was not involved in this evaluation may generate snippets that are relevant as well but not included in the assessments, because none of the participating systems returned it in the output. When only three systems are participating, this is very likely to happen. This is illustrated in the following example:

For a given topic (i.e. topic 2) the topic description is as follows:

I want to explain to the people about bird flu and its symptoms occur in human. I assume that the people have no knowledge at all about it.

Below a selection of relevant snippets is given, from the output of a system that uses paragraphs and a ranking mechanism based upon candidate snippets with IDF-only, which are not part of the assessments and therefore not delivered by the participating systems. Since this problem often occurs we may conclude that only three participants is not enough for this kind of evaluation.
Usually, "avian influenza virus" refers to influenza A viruses found chiefly in birds, but infections with these viruses can occur in humans. The risk from avian influenza is generally low to most people, because the viruses do not usually infect humans. However, confirmed cases of human infection from several subtypes of avian influenza infection have been reported since 1997. Most cases of avian influenza infection in humans have resulted from contact with infected poultry (e.g., domesticated chicken, ducks, and turkeys) or surfaces contaminated with secretion/excretions from infected birds. The spread of avian influenza viruses from one ill person to another has been reported very rarely, and has been limited, inefficient and unsustainable.

Birds, just like people, get the flu. Bird flu viruses infect birds, including chickens, other poultry and wild birds such as ducks. Most bird flu viruses can only infect other birds. However, bird flu can pose health risks to people. The first case of a bird flu virus infecting a person directly, H5N1, was in Hong Kong in 1997. Since then, the bird flu virus has spread to birds in countries in Asia, Africa and Europe.

Furthermore for some of the topics, the assessments are not carefully created, as mentioned before in the evaluation section. Therefore the dataset is not usable for evaluation. In the future this problem can be solved by carefully making the assessments for the upcoming conference (i.e. WebCLEF 2008).

Unfortunately the problem with the evaluation method is more complex and a solution is not directly available, although there has been done a lot of research at this problem already. For designing a new evaluation method many choices have to be made. Whatever these choices will be, the goal must be offering system developers much insight into its parameters. This is already difficult enough, but in addition the performance of these systems is far from high. The combination of these facts makes it hard to develop task-oriented evaluations that are both related to the researchers’ interests and are not too far beyond their systems’ capabilities [21].

In the WebCLEF task, text quality (i.e. can the system produce ‘proper’ sentences and ‘properly connected’ discourse) is not that important, since only extractive approaches are used so far. However concept capturing (i.e. does the summary capture the key concepts of the sources) is more important. Unfortunately it is much harder to answer the second question, because not only it involves judgments about conceptual importance in the source but because concepts, especially complex relational ones, are not clear cut and they may be variably expressed [21].

One of the problems, mentioned earlier, is that the assessments do not contain enough relevant snippets and therefore are not reusable. This is caused by the fact that the assessments are based upon a pool of snippets that is not large enough due to the small number of participating systems. [22] showed that for this reason measures based on recall are highly uncertain. There has been done some research to cope with incomplete assessments (e.g. bpref [23]). This can be a possible solution for the reusability problem. TRECEVAL, for example is a program that also reports this bpref value. Unfortunately TRECEVAL cannot be used for the WebCLEF evaluation, since it is impossible to give snippets an id. This in turn, is caused by the fact that the participating systems have to extract the snippets from the document collection itself and therefore not every system extracts the same snippets.

This leaves us with the problem of how to evaluate in general. An approach that is close to the current evaluation method, and therefore a reasonable solution, is providing the offsets (i.e. the start and end of a passage in the document) of the delivered snippets. When this information would be available then simply the amount of overlap can be calculated to get an indication of the performance. A similar approach is already used in XML Retrieval [24].

Another more common approach for evaluating extractive summaries, which is the case in WebCLEF, is automatic comparison between reference and system summaries using n-grams. Originally this approach is applied to machine translation, but it has been developed in the ROUGE program for summary evaluation as well [25].

6. CONCLUSION & FUTURE WORK

In this paper different methods for extracting snippets, ranking snippets and measuring similarity are described. Unfortunately it was not possible to compare the performance of these methods with each other due to a wrong evaluation method and dataset of WebCLEF 2007.

To avoid this problem in the future some possible solutions are discussed in the discussion section. In the near future (i.e. WebCLEF 2008) there will be a new dataset consisting of 61 multilingual topics. This dataset will hopefully be assessed more carefully to prevent the problems already discussed in this paper. To solve the problems with the evaluation method itself, another evaluation method has to be used. This all is up to the organizers of WebCLEF 2008.

The system described in this paper will participate as well in WebCLEF 2008. According to these results more insight can be given in the performance of the different methods used by the system. Unfortunately this will not be in time for the Bachelor report.

ACKNOWLEDGEMENTS

I would like to thank my supervisors: Claudia Hauff, Djordje Hiemstra, Franciska de Jong and Dolf Trieschnigg for their valuable feedback. I also would like to thank Valentijn Jikoun and Maarten de Rijke for providing me with their system.

REFERENCES


