Using Centrality to Rank Web Snippets

Valentin Jijkoun and Maarten de Rijke

ISLA, University of Amsterdam
jijkoun,mdr@science.uva.nl

Abstract. We describe our participation in the WebCLEF 2007 task, targeted at snippet retrieval from web data. Our system ranks snippets based on a simple similarity-based centrality, inspired by the web page ranking algorithms. We experimented with retrieval units (sentences and paragraphs) and with the similarity functions used for centrality computations (word overlap and cosine similarity). We found that using paragraphs with the cosine similarity function shows the best performance with precision around 20% and recall around 25% according to human assessments of the first 7,000 bytes of responses for individual topics.

1 Introduction

The WebCLEF 2007 task[1] differed substantially from the previous editions (2005–2006 of WebCLEF). Rather than retrieving a ranked list of web documents relevant to a topic, in the 2007 setup, systems were asked to return a ranked list of *snippets* (character spans) extracted from the top 1,000 web documents identified using the Google web search engine. The definition of the retrieval unit (snippet) was left up to a system, and thus the task is targeting *information retrieval* rather than *document retrieval*.

The remainder of this paper is organized as follows. We describe the WebCLEF 2007 task and topics in Section 2, present the architecture of our system in Section 3, describe our three runs, evaluation measures and evaluation results in Section 4, and conclude in Section 5.

2 Task and Topics

In WebCLEF 2007 for each topic systems are provided with the following information:

- topic title (e.g., *Big Bang Theory*);
- description of the information need (e.g., *I have to make a presentation about Big Bang Theory for undergraduate students. I assume that the students have some basic knowledge of physics.*);
- languages in which information can be returned;

[1] URL: http://ilps.science.uva.nl/WebCLEF/WebCLEF2007

– known sources: URLs and content of pages already "known" to the topic author;
– a list of web pages (original and text format) retrieved using Google with queries provided by the topic author (e.g., *Big Bang*); for each query, at most 1,000 pages are included in the list.

The task of a system is to return a ranked list of text spans from the provided web pages that, together, would satisfy the user's information need.

Task organizers provided two development topics and 30 test topics.

3 System Architecture

For each topic, our system used only text versions of the web documents. On the one hand, the decision not to use the original versions of the documents (HTML, PDF, Postscript, etc.) led to some noise in the output of the system. In the text versions, the text encoding was often broken, which was especially problematic for non-English documents (the task included Spanish and Dutch topics and pages). Moreover, in cases where an original document was a double-column PDF, in the corresponding text version, the lines of the columns were often intervened, making the text version hardly readable for humans. For some of the original documents (e.g., for Word files) text versions were missing, and therefore our system did not use these documents at all. On the other hand, using only text version simplified the data processing considerably:

– no sophisticated content extraction had to be developed, and
– the text versions often preserved some text layout of the original pages (e.g., paragraph starts), which we used to detect suitable snippet boundaries.

Given a topic, our system first identifies *candidate snippets* in the source documents by simply splitting the text of the documents into sentences (using punctuation marks as separators) or into paragraphs (using empty lines as separators). The same snippet extraction method is applied to the text of the "known" pages for the topic, resulting in a list of *known snippets*. We ignored candidate and known snippets shorter that 30 bytes.

3.1 Ranking snippets

We rank the candidate snippets based on *similiarity-based centrality*, which is a simplified version of the graph-based snippet ranking of [2], inspired by the methods for computing authority of the Web pages, such as PageRank and HITS [4]. For each candidate snippet we compute a *centrality score* by summing similarities of the snippet with all other candidate snippets. Then, to avoid assigning high scores to snippets containing information that is already known to the user, we subtract from the resulting centrality score similarities of the candidate snippet with all known snippets. As a final step, we remove from consideration candidate snippets whose similarity to one of the known snippets is higher than a threshold. The pseudocode for this calculation is shown below:

let $c_1 \dots c_n$ be candidate snippets
let $k_1 \dots k_m$ be known snippets
for each candidate snippet c
    let $score(c) = 0$
    for each candidate snippet c'
        let $score(c) = score(c) + sim(c, c')$
    for each known snippet k
        let $score(c) = score(c) - sim(c, k)$
    for each known snippet k
        if $sim(c, k) > sim_{max}$
            let $score(c) = 0$

Finally, the candidate snippets are ranked according to $score(\cdot)$ and top snippets are returned so that the total size of the response is not larger that 10,000 bytes.

3.2 Similarity between snippets

A key component of our snippet ranking method is the snippet similarity function $sim(x, y)$. Similarly to [3], we conducted experiments with two versions of the similarity function: one based on word overlap and one based on the cosine similarity in the vector space retrieval model. Specifically, for two text snippets, *word overlap* similarity is defined using the standard Jaccard coefficient on snippets considered as sets of terms:

$$sim_{wo}(x, y) = \frac{|x' \cap y'|}{|x' \cup y'|},$$

where x' and y' are sets of non-stopwords of snippets x and y respectively.

The *vector space* similarity between two snippets is defined as the cosine of the angle between the vector representations of the snippets computed using the standard TF.IDF weighting scheme:

$$sim_{vs}(x, y) = \frac{\overrightarrow{x} \cdot \overrightarrow{y}}{\sqrt{\overrightarrow{x} \cdot \overrightarrow{x}}\sqrt{\overrightarrow{y} \cdot \overrightarrow{y}}}.$$

Here, $\overrightarrow{a} \cdot \overrightarrow{b}$ denotes the scalar product of vectors $\overrightarrow{a}$ and $\overrightarrow{b}$.

Components of the vectors correspond to distinct non-stopword terms occuring in the set of candidate snippets. For a term t, the value of the component $\overrightarrow{a}$ is defined according to the TF.IDF weighting scheme:

$$\overrightarrow{a}(t) = TF(a, t) \cdot \log\left(\frac{n}{|\{c_i : t \in c_i\}|}\right).$$

Here, $TF(a, t)$ is the frequency of term t in snippet a and $c_1, \dots, c_n$ are all candidate snippets.

Both versions of the similarity function produce values between 0 and 1. The similarity threshold sim_{max} for detecting near-duplicates is selected based on manual assessment of duplicates among candidate snippets for the development topics.

4 Submitted Runs and Evaluation Results

Our goal was to experiment with the units of snippet retrieval and with similarity functions. We submitted three runs:

- **UvA sent wo** – snippets defined as sentences, word overlap used for ranking;
- **UvA par wo** – snippets defined as paragraphs, word overlap for ranking;
- **UvA par vs** – paragraphs, vector space similarity.

The evaluation measures used for the task were character-based precision and recall, based on human assessments of the first 7,000 bytes of system's response. *Precision* is defined as the length of the character spans in the response identified by humans as relevant, divided by the total of the response (limited to 7,000 characters). *Recall* is defined as the length of spans in the reponse identified as relevant, divided by the total length of all distinct spans identified as relevant for the responses submitted by all systems.

The evaluation results for the three runs are shown below:

Run	Precision	Recall
UvA sent wo	0.0893	0.1133
UvA par wo	0.1959	0.2486
UvA par vs	**0.2018**	**0.2561**

The results indicate that paragraphs provide better retrieval units and using a more sophisticated similarity function based on the vector space model has a slight positive effect on the performance. Unfortunately, we did not have enough time to analyse performance of the versions of the system per topic or to check whether the improvement with the vector space similarity function is significant.

Overall, we believe that the paragraph-based runs may serve as a reasonable baseline for the WebCLEF task: around 1/5 of the returned character content is considered relevant by human assessors. At the same time, such performance is probably not sufficient for a real-life information retrieval system.

5 Conclusions

We have described our participation in the WebCLEF 2007 snippet retrieval task. In our submission we experimented with retrieval units (sentences vs. paragraphs) and with similarity functions used for semantic centrality computations (word overlap vs. cosine similarity). We found what using paragraphs with the cosine similarity function shows the best performance with precision around 20% and recall around 25% according to human assessments of the first 7,000 bytes of per-topic responses.

Detailed analysis of the performance of the runs is part of our immediate agenda for future work. Another interesting direction for further study is the similarity model suitable for short snippets. The vector space model that we use in this paper is not necessarily the best option. However, it has been shown (see, e.g., [1, 2]) that more sophisticated models do not necessarily lead to improvements when working with short text fragments.

Acknowledgements

The research presented in this paper was supported by NWO under project numbers 017.001.190, 220.80.001, 264.70.050, 354.20.005, 600.065.120, 612.13.001, 612.000.106, 612.066.302, 612.069.006, 640.001.501, and 640.002.501. We are grateful to all participants and assessors of the WebCLEF 2007 task.

References

1. J. Allan, C. Wade, and A. Bolivar. Retrieval and novelty detection at the sentence level. In *SIGIR '03*, pages 314–321, 2003.
2. Sisay Fissaha Adafre, Valentin Jijkouni, and Maarten de Rijke. Fact discovery in Wikipedia. In *IEEE/WIC/ACM International Conference on Web Intelligence, 2007*, 2007.
3. Valentin Jijkoun and Maarten de Rijke. Recognizing textual entailment: Is lexical similarity enough? In I. Dagan, F. Dalche, J. Quinonero Candela, and B. Magnini, editors, *Evaluating Predictive Uncertainty, Textual Entailment and Object Recognition Systems*, volume 3944 of *LNAI*, pages 449–460. Springer, 2006.
4. Bing Liu. *Web Data Mining. Exploring Hyperlinks, Contents and Usage Data*. Springer, 2006.