Snip!

Andrew Trotman, Matt Crane

Department of Computer Science
University of Otago
Dunedin, New Zealand

Abstract. The University of Otago submitted runs to the Snippet Retrieval Track and the Relevance Feedback tracks at INEX 2011. Snippets were generated using vector space ranking functions, taking into account or ignoring structural hints, and using word clouds. We found that using passages made better snippets than XML elements and that word clouds make bad snippets. In our runs in the Relevance Feedback track we were testing the INEX gateway to C/C++ and blind relevance feedback (with and without stemming). We found that blind relevance feedback with stemming does improve prevision in the INEX framework.

Keywords: Wikipedia, Snippet Generation, Procrastination.

1 Introduction

In 2011 the University of Otago participated in two tracks it had not previously experiment with: the Snippet Retrieval Track and the Relevance Feedback Track. Six snippet runs were submitted and three relevance feedback runs were submitted. This contribution discusses those runs.

For details of the INEX document collection and the "rules" for the tracks the interested reader is referred to the track overview papers in this volume.

2 Snippets

2.1 Runs

A total of six runs were submitted:

First-p: in this run the snippet was the first 300 characters of the first <p> element in the document. This run was motivated by the observation that the start of a Wikipedia article typically contains an overview of the document and is therefore a good overview of the paper. In this run and all submitted runs the snippet was constructed so that it always started at the beginning of a word and ended at the end of a word and all XML tag content was discarded.

Top-tf-paragraph: in this run the snippet was the first 300 characters from the paragraph (<p>) element with the highest sum of query term occurrences (that is, for a two word query it is sum of the tf's of each term).

Top-tf-passage : in this run the snippet was the 300 character (word aligned) sliding window with the highest sum of query term occurrences. Since there are usually many such possible windows, the window was centered so that the distance from the start of the window to the first occurrence was the same (within rounding errors) as the distance from the last occurrence to the end of the window (measured in bytes).

These three runs together form experiment 1 in which the aim is to determine whether a snippet is better formed from an element or a passage.

Top-tficf-paragraph: in this run the snippet was the first 300 characters of the paragraph with the highest *tf* * *icf* weight were $icf_t = \log(C/c_t)$ were C is the length of the collection (in term occurrences) and c is the number of times term t occurs.

Top-tficf-passage: in this run the snippet was the first 300 character (word aligned) sliding window with the highest tf * icf score.

The tficf runs along with the tf runs form experiment 2 in which the aim is to determine the most effective way of choosing a passage or paragraph.

The final run was

KL: in this run the KL-divergence between each term in the document and the collection was used to order all terms in the document. From this ordering the top n were chosen as the snippet so that the snippet did not exceed 300 characters.

This final run forms experiment 3 in which the aim is to determine whether snippets are better form using extractive techniques (phrases) are better than summative techniques (word clouds).

2.2 Results

The preliminary results against the GM metric are presented in Table 1. From this table it can be seen that the passage runs were universally better than the element runs and that tf.icf runs were universally better than tf runs. Our best run was passage-based tf.icf.

No run performed as well as the QUT run that simply took the first 300 characters from the document. Our run first-p took the first paragraph element from the document, but this is not equivalent. Our run did not include the document title, and in some documents the first paragraph was empty. We believe that this demonstrates the importance of putting the title into the snippet (context matters).

Other participants also submitted runs generated as word clouds, and those runs also performed below the median, suggesting that word clouds do not make good snippets.

Table 1: Snippet Track results for University of Otago runs

Rank (of 41)	**Run**	**GM**
1	LDKE-1111	0.5705
4	QUTFirst300	0.5416
11	top_tficf_passage	0.5242
27	top_tf_passsage	0.4648
28	top_tf_p	0.4574
34	top_tficf_p	0.4337
37	first_p	0.4044
39	kl	0.3598

2.3 Observations from assessing

In total 6 topics were assessed by participants at the University of Otago. A post-assessment debriefing by the four assessors resulted in the following observations:

Snippets that included the title of the document were easier to assess than those that did not. It is subsequently predicted that those runs will generally score better than runs that did not. A recommendation is made to the track chairs to either automatically include the document title in the assessment tool, or to make it clear that the snippet may include the document title.

Snippets that were extractive from multiple parts of the document (included ellipses) generally contained multiple snippets each of which was too short to be useful and collectively not any better.

Snippets made from word clouds were generally instantly dismissible as up-helpful.

Snippets that contained what appeared to be the section / subsection "path" through the document generally took so much space that the remaining space for the extractive snippet was too short for a useful snippet.

2.4 Further work

If the track is run in 2012 then from the observations it would be reasonable to submit a run that contains the document title, a single snippet extracted from the document, and the title of the section from which the snippet was extracted. The method of extraction is unclear and would depend on the results of the experiments submitted to this track in 2011.

3 Relevance Feedback

The purpose of the Otago relevance feedback runs was twofold: The first purpose was to experiment with the INEX relevance feedback infrastructure. In particular, the infrastructure was written in Java but the search engine Otago uses (ATIRE) is written in C++. The gateway from Java to C++ was provided by INEX.

The second purpose was to determine whether or not blind relevance feedback is an effective method of improving whole-document search results on Wikipedia. To this end the runs submitted by Otago ignored the human assessments and only returned whole documents.

3.1 Runs

A total of three runs were submitted:

BM25: in this run the documents are ranked using BM25 (k1=0.9, b=0.4). No relevance feedback was performed. This runs forms an out-of-the-box baseline. Is it the result of running the untrained ATIRE search engine over the documents.

BM25-RF: in this run the documents are ranked using BM25 (as above), then from the top 5 document the top 8 terms were selected using KL-divergence. These were then added to the query (according to Rocchio's algorithm) and BM25 was again used to get the top results. Terms added to the query had an equal weight to those already there, and terms already in the query could be added. The parameters 5 and 8 were chosen through learning over the training data.

BM25-RF-S: in this run the documents are ranked using BM25, then from the top 5 document the top 8 terms were selected using KL-divergence (as above). Additionally the S-stemmer was used in the initial and second query. Additional to this the parameters for BM25 were learned using a grid search (k1=0.5 b=0.5). Again training was on the INEX supplied training data.

In all runs blind relevance feedback was used and the user's assessments were ignored. As such these runs form a good baseline for ignoring the user.

3.2 Results

A subset of the official INEX published results are presented in Table 1 and the Precision / Recall graph is presented in Figure 1. The focused retrieval results are not presented as whole-document retrieval was used.

From the results, it appears as though relevance feedback has no effect on the performance of the search engine, but stemming does. It is already know that stemming works on the INEX Wikipedia collection, but unexpected that Rocchio Feedback does not. This result needs to be verified as it could be a problem with the run or a problem with the assessment method.

Table 2: INEX Published Results (from INEX)

Precision	Reference	Reference no feedback	Otago BM25	Otago BM25-RF	Otago BM25-RF-S
P@5	0.500	0.500	0.540	0.540	0.520
P@10	0.515	0.435	0.445	0.445	0.485
P@15	0.490	0.400	0.417	0.417	0.450
P@20	0.468	0.385	0.398	0.398	0.398
R-Precision	0.413	0.336	0.360	0.360	0.357

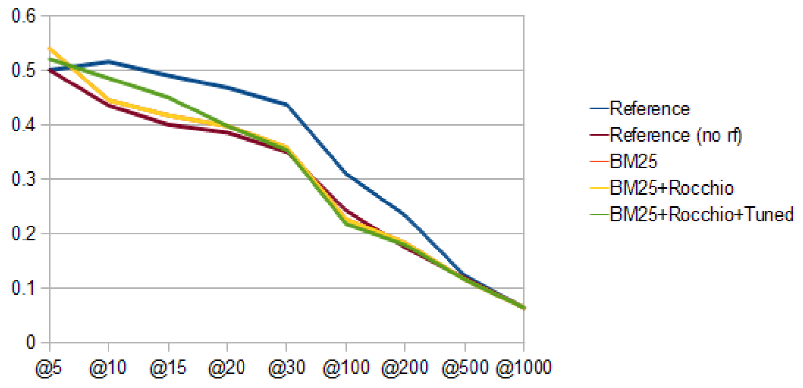

Figure 1: Official INEX Relevance Feedback Result (from INEX)

3.4 Further work

If the track is run in 2012 then it is reasonable to build on the baseline by including the user's assessments in the run. This could be done by performing a process similar to run BM25-RF-S at each assessment point and returning the top as-to un-seen document.

However, before any further work is done it is important to understand why relevance feedback does not appear to have an effect on this collection.

4. Conclusions

The University of Otago submitted six snippet runs and three feedback runs. These runs form baselines for experiments in improving the quality of the results in the search engine.

It is not clear why the relevance feedback method had no effect on precision. In further work this will be investigated.

References

The interested reader is referred to the overview papers of INEX 2011, especially the overview of the snippet and relevance feedback tracks.