Wikipedia *Ad hoc* Passage Retrieval and Wikipedia Document Linking

Dylan Jenkinson and Andrew Trotman

Department of Computer Science
University of Otago
Dunedin
New Zealand
{djenkins, andrew}@cs.otago.ac.nz

Abstract. *Ad hoc* passage retrieval within the Wikipedia is examined in the context of INEX 2007. An analysis of the INEX 2006 assessments suggests that fixed sized window of about 300 terms is consistently seen and that this might be a good retrieval strategy. In runs submitted to INEX, potentially relevant documents were identified using BM25 (trained on INEX 2006 data). For each potentially relevant document the location of every search term was identified and the center (mean) located. A fixed sized window was then centered on this location. A method of removing outliers was examined in which all terms occurring outside one standard deviation of the center were considered outliers and the center recomputed without them. Both techniques were examined with and without stemming.

For Wikipedia linking we identified terms within the document that were over-represented and from the top few generated queries of different lengths. A BM25 ranking search engine was used to identify potentially relevant documents. Links from the source document to the potentially relevant documents (and back) were constructed (at a granularity of whole document). The best performing run used the 4 most over-represented search terms to retrieve 200 documents, and the next 4 to retrieve 50 more.

1. Introduction

The University of Otago participated in new tasks introduced to INEX in 2007. In the passage retrieval task three runs were submitted to each of the focused, relevant-in-context and best-in-contest tasks (and a fourth run was not submitted). In the Link-the-Wiki track five runs were submitted. In all cases performance was adequate (average or better).

An analysis of the 2006 INEX assessments (topics version:2006-004, assessments version:v5) shows that documents typically contain only one relevant passage, and that that passage is 301 characters in length. This leads to a potential retrieval strategy of first identifying potentially relevant documents, then from those identifying the one potentially relevant passage (of a fixed length). In essence this has

reduced the passage retrieval problem to that of placing a fixed sized window on the text.

The approach we took was to identify each and every occurrence of each search term within the document. From there the mean position was computed and the window centered there. Outliers could potentially affect the placement of the window so an outlier reduction strategy was employed. All occurrences lying outside one standard deviation of the mean were eliminated and the mean recomputed. This new mean was used to place the window.

Porter stemming [6] was tested in combination with and without outlier reduction. Of interest to XML-IR is that our approach does not use document structure to identify relevant content. Kamps & Koolen [4] suggest relevant passages typically start (and end) on tag boundaries, however we leave exploitation of this to future work.

Our best passage retrieval runs when compared to element retrieval runs of other participants ranked favorably.

In the Link-the-Wiki task we again ignored the document structure and used a naive method. A score for each term in the orphaned document was computed as the ratio of length normalized document frequency to the expected frequency computed from collection statistics. Terms were ranked then queries of varying length (from 1 to 5 terms) were constructed from the top ranked terms in the list.

No attempt was made to identify anchor text or best entry points into target documents – instead linking from document to document was examined. We found that in this kind of linking query lengths of 4 terms performed best.

2. *Ad hoc* Passage Retrieval

The INEX evaluation forum currently investigates subdocument (focused) information retrieval in structured documents, specifically XML documents. Focused retrieval has recently been defined as including element retrieval, passage retrieval and question answering [11]. In previous years INEX examined only element retrieval but in 2007 this was extended to include passage retrieval and book page retrieval. Common to all these paradigms is the requirement to return (to the user) only those parts of a document that are relevant, and not the whole document.

These focused searching paradigms are essentially identical and can be compared on an equal basis (using the same queries and metrics). If an XML element is specified using the start and end word number within a document (instead of XPath) then an XML element can be considered a passage. The same principle is true of a book page if word numbers are used instead of page numbers. A question answer within the text can also be considered a passage if it, too, is consecutive in the text.

Our interest in passage retrieval is motivated by a desire to reduce the quantity of irrelevant text in an answer presented to a user, that is, to increase focused precision. We believe that element granularity is too coarse and that users will necessarily be presented with irrelevant text along with their answers because any element large enough to fully contain a relevant answer is also likely to be sufficiently large that it contains some irrelevant text. Exactly this was examined by Kamps & Koolen [4]

who report that, indeed, the smallest element that fully contains a relevant passage of text often contains some non-relevant text. The one way to increase precision is to remove the irrelevant text from the element, and one obvious way to do this is to shift to a finer granularity than element, perhaps paragraph, sentence, word, or simply passage.

2.1. INEX 2007 Tasks

There were three distinct retrieval tasks specified at INEX 2007: focused retrieval; relevant-in-context retrieval; and best-in-context retrieval. In focused retrieval the search engine must generate a ranked non-overlapping list of relevant items. This task might be used to extract relevant elements from news articles for multi-document summarization (information aggregation).

The relevant-in-context task is user-centered, and the aim is to build a search engine that presents, to a user, a relevant document with the relevant parts of that document highlighted. For evaluation purposes documents are first ranked on topical relevance then within the document the relevant parts of the document are listed.

Assuming a user can only start reading a document from a single point within a document, a search engine should, perhaps, identify that point. This is the aim of the best-in-context task, to rank documents on topical relevance and then for each document to identify the point from which a user should start reading in order to satisfy their information need.

For all three tasks both element retrieval and passage retrieval are applicable. For both it is necessary to identify relevant documents and relevant text within those documents. For element retrieval it is further necessary to identify the correct granularity of element to return to the user (for example, paragraph, sub-section, section, or document). For passage retrieval it is necessary to identify the start and end of the relevant text. It is not yet known which task is hardest, or whether structure helps in the identification of relevant text within a document. It is known that the precision of a passage retrieval system must, at worst, be at least equal to that of an element retrieval system.

2.2. Passage Retrieval

Passages might be specified in several different ways: an XML element, a start and end word position, or any granularity in-between (sentences, words, and so on). The length of a passage can be either fixed or variable. Within a document separate passages might either overlap or be disjoint.

If element retrieval and passage retrieval are to be compared on an equal basis it must be possible to specify an XML element as a passage. This necessitates a task definition that allows variable sized passages. Interactive XML-IR experiments show that users do not want overlapping results [10], necessitating a definition of disjoint passages. The INEX passage retrieval tasks, therefore, specify variable length non-overlapping passages that start and end on word boundaries. We additionally chose to

ignore document structure as we are also interested in whether document structure helps with the identification of relevant material or not.

2.3. Window Size

Previous experiments suggest that fixed sized windows of between 200 and 300 words is effective [2]. To determine the optimal size for the Wikipedia collection an analysis of the INEX 2006 results was performed.

In 2006 INEX participants assessed documents using a yellow-highlighting method that identified all relevant passages within a document. For each passage the start and end location are given in XPath and the length is given in characters. Best entry points are also specified.

Kamps & Koolen [4] performed a thorough analysis of the assessments and report a plethora of statistics. We reproduce some of those analyses, but present results in a different way.

Figure 1 presents the number of relevant documents in the assessment set that contain the given number of passages. The vast majority of relevant documents (70.63%) contain only one relevant document. This suggests that any passage retrieval algorithm that chooses to identify only one relevant passage per document will be correct the majority of the time. Because it is reasonable to expect only one relevant passage per document the tasks can be simplified to identifying *the relevant passage* in a document, not the relevant *passages* within a document. 17.27% contain 2 passages and 12.10% contain 3 or more passages.

Figure 2 presents the mean passage length (in words) of a passage as the number of passages within a document increases. It was reasonable to expect that as the number of passages increased that the mean length of the passage would decrease as there is a natural limit on the sum of the lengths (the document length). Instead it can be seen that the average length is about constant. In a multiple-assessor experiment on the same document collection Trotman *et al*. [12] asked assessors whether they preferred to identify fixed-sized passages or variable sized passages and found that half preferred fixed sized passages of about a paragraph in length. This is consistent with the observation that passages are all about the same length – when a single passage is seen the mean is 283 words, but if more than one passage is sent then it varies between 73 and 153 words. Given this is the case then it is reasonable to expect that the length of a document is related to the number of passages it contains – this is shown to be the case in Figure 3 where it can be seen that document length increases with the number of passages.

The mean relevant content per document is 301 words. In Figure 4 the length of all relevant passages in all documents is presented – very few passages are long (over 1000 words) or short (under 10 words).

Given the mean length of relevant content in a document is about 300 words, and that only one passage is expected per document, it is reasonable to develop a passage retrieval algorithm that identifies one passage of 300 words. There does, however, remain the problem of identifying where, within a document, that passage should be placed.

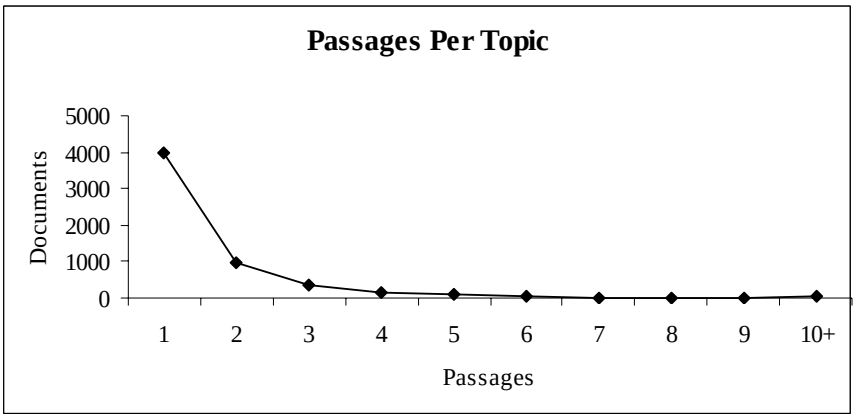

Figure 1: Number of documents containing the given number of passages.

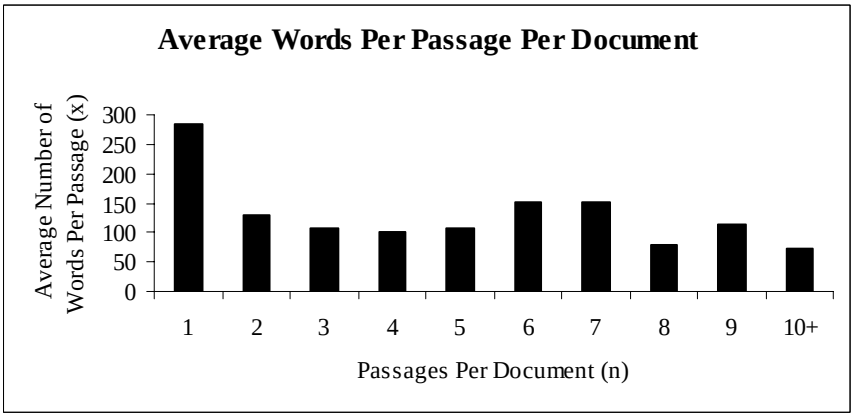

Figure 2: Passage length varies with number of passages per document.

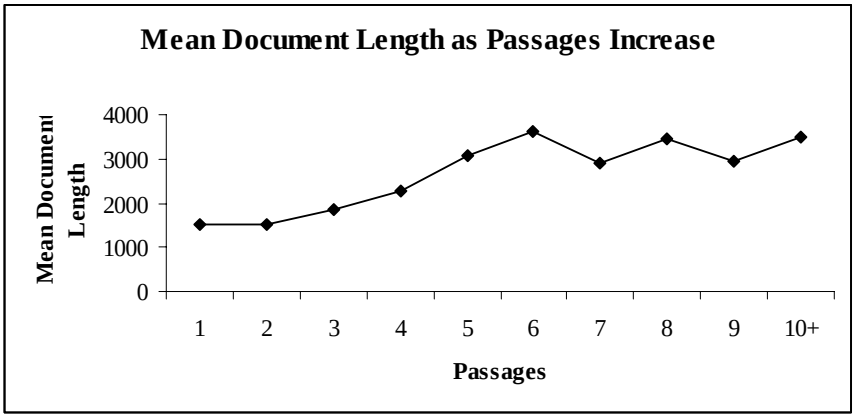

Figure 3: Mean document length as the number of passages increases.

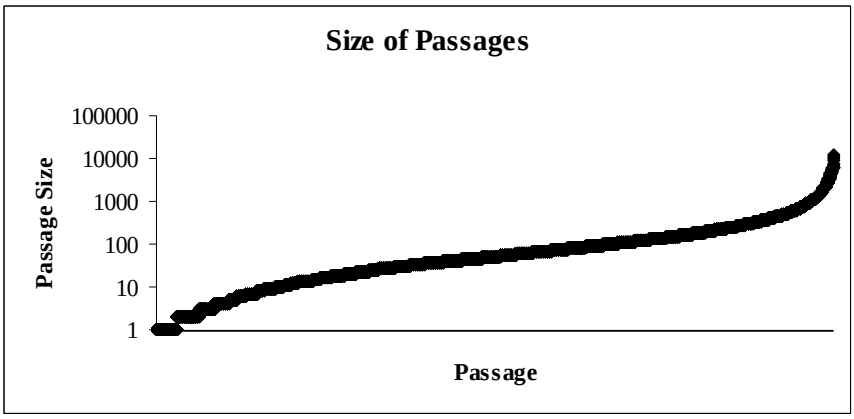

Figure 4: Log of passage size for all relevant passages.

2.4. Window Location

A heat map of the document can be built by noting the location of all search terms within the document. Areas where search terms do not occur (cold areas) are unlikely to be relevant to the user's query; conversely areas where there are many occurrences of the search terms (hot areas) are likely to be relevant.

Our hypothesis is that centering the one fixed-sized window over the middle of the dense areas will be an effective retrieval strategy. This method ignores the structure of the document, which we believe makes the comparison to element-retrieval systems of particular interest.

For each document identified as potentially relevant the XML structure is removed and the location of all occurrences of all search terms is identified. The mean of these locations is considered to be the center of relevance and so the window is centered on this point. If the window extends outside the document (before the beginning for example) then the window is truncated at the document boundary.

Problematically, in a well structured document it is reasonable to assume search terms will occur in the abstract and conclusions, but for the relevant text to occur elsewhere, in the body of the document for example. Several early or late term occurrences might shift the window towards the outliers which will in turn reduce precision. A method is needed to identify and remove outliers before the window is placed. We hypothesize that removing outliers will increase precision.

Two window placement methods were implemented: *meanselection* and *stddevselection*. With *meanselection* the center point (mean) of all occurrences of all search terms was used. With *stddevselection* the mean search term position was found and the standard-deviation computed. Then all occurrences outside one standard deviation from the mean were discarded. A new mean was then computed from the pruned list, and this was used as the passage midpoint.

2.5. Stemming

The identification of search terms within the document is essential to the performance of the window placement technique. It is reasonable to expect authors to use different morphological variants and synonyms of search terms within their documents. The inclusion of these in the algorithms is, therefore, important. We experimented with Porter's stemming algorithm [6].

2.6. Potentially Relevant Documents

The identification of relevant documents in *ad hoc* retrieval has been studied extensively by others. Several effective methods have been presented including language models [13], pivoted cosine normalization [9], and BM25 [7]. We chose BM25.

BM25 is parametric and requires scores for *k1*, *k3* and *b*. We used genetic algorithms [1] and trained on the INEX 2006 data to obtain good scores. The details are not important and we just report that the training resulted in the values 0.487, 25873, and 0.288 for *k1*, *k3* and *b* respectively.

Stemming was not used during training and was not used to identify potentially relevant documents

2.7. Best Entry Points

Kamps *et al*. [5] show a correlation between the best entry point and the start of the first relevant passage. They report 67.6% of best entry points in a single-passage document lying at the start of the passage (17.16% before and 15.24% after). For a document with two passages these numbers are substantially different. The chance that the best entry point coincides with the start of the first passage in the document is reduced to 35.33%, whilst the chance that the best entry point is before the first passage is increased to 45.21%. The chance of the best entry point coming after the first passage is about 19.46%. Figure 5 presents our analysis. It shows, for all documents with a single relevant passage, the distance (in characters) from the start of that passage to the best entry point. The vast majority of all passages start at or very close to the best entry point. This suggests a best entry point identification strategy of "just choose the start of the first relevant passage".

3. *Ad Hoc* Experiments

3.1. *Ad Hoc* Runs

We conducted two experiments: the first was the effect of stemming, the second was the effect of removing outliers. This gave 4 possible combinations (runs) for each task as outlined in Table 1. However, as we were only permitted to submit 3 official runs per task and so the last run was scored informally. We expect the performance with standard-deviation and stemming to be most effective as this run will be better at identifying occurrences of search terms, while also better at removing outliers.

The same runs were submitted to each of the *ad hoc* tasks (focused, relevant-in-context, and best-in-context) and the runs differ only in name.

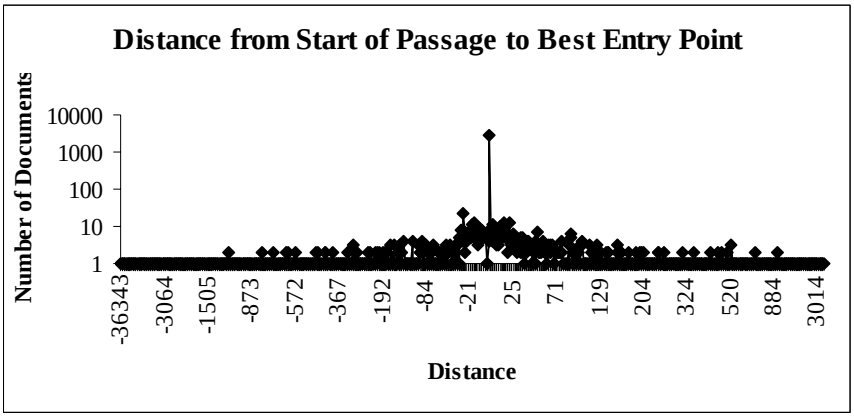

Figure 5: Distance (in characters) of the best entry points from the start of the first passage. Negative are before the first passage.

Table 1. Runs submitted to the INEX 2007 *ad hoc* track.

Run	Focused	Relevant-in-context	Best-in-context
1	DocsNostem-PassagesStem-StdDevYes-Focused	DocsNostem-PassagesStem-StdDevYes	DocsNostem-PassagesStem-StdDevYes-BEP
2	DocsNostem-PassagesStem-StdDevNo-Focused	DocsNostem-PassagesStem-StdDevNo	DocsNostem-PassagesStem-StdDevNo-BEP
3	DocsNostem-PassagesNoStem-StdDevNo-Focused	DocsNostem-PassagesNoStem-StdDevNo	DocsNostem-PassagesNoStem-StdDevNo-BEP
4	DocsNostem-PassagesNoStem-StdDevYes-Focused	DocsNostem-PassagesNoStem-StdDevYes	DocsNostem-PassagesNoStem-StdDevYes-BEP

3.2. *Ad hoc* Results

Table 2 presents the scores and relative rank of the focused runs. The best run used stemming but not the *stddevselection* method. The relative rank of all runs is similar and the differences are small.

Of particular note is that of the 79 runs submitted to the task our runs that did not use document structure performed adequately (in the top 33%).

In the tables in this section column 3, marked +, represents scored computed at the University of Otago using the released INEX evaluation software whereas column 2 represents the official score released on the INEX website (so the score for the fourth run is not given).

Table 2. Focused task results computes at 0.01 recall. $^{+}$values computed locally.

Run	iMAP	iMAP^{+}	Rank
DocsNostem-PassagesStem-StdDevYes-Focused	0.4659	0.4609	30
DocsNostem-PassagesStem-StdDevNo-Focused	0.4716	0.4698	26
DocsNostem-PassagesNoStem-StdDevYes-Focused	-	0.4645	-
DocsNostem-PassagesNoStem-StdDevNo-Focused	0.4705	0.4688	28

The performance of the runs submitted to the relevant-in-context task is shown in Table 3. Here there is no material difference in the score of the runs. Of 66 runs submitted to the task our top run that ignores structure performed averagely (32nd).

Table 3. Relevant-in-context results. $^{+}$values computed locally.

Run	MAgP	MAgP^{+}	Rank
DocsNostem-PassagesStem-StdDevYes	0.1028	0.1010	33
DocsNostem-PassagesStem-StdDevNo	0.1021	0.1014	34
DocsNostem-PassagesNoStem-StdDevNo	0.1033	0.1020	32
DocsNostem-PassagesNoStem-StdDevYes	-	0.1012	-

The performance with respect to the best-in-context task is shown in Table 4. Here outlier reduction was effective but stemming was not. The relative system performance of our best submitted run was 42 of 71.

Table 4. Best-in-context results. . $^{+}$values computed locally.

Run	MAgP	MAgP^{+}	Rank
DocsNostem-PassagesNoStem-StdDevYes-BEP	-	0.1101	-
DocsNostem-PassagesStem-StdDevYes-BEP	0.1061	0.1083	43
DocsNostem-PassagesStem-StdDevNo-BEP	0.1064	0.1066	42
DocsNostem-PassagesNoStem-StdDevNo-BEP	0.1060	0.1062	44

3.3. Discussion

We chose to ignore document structure and submitted run that, instead, simply used term locations to place a fixed sized window on the text. From the relative system performance it is reasonable to conclude that selecting a single fixed sized passage of text produces reasonable results.

The stemming experiment shows that stemming is not important for choosing the location of the window. When searching a very large document collection it is reasonable to ignore stemming because any relevant document will satisfy the user's information need. This should not be the case when looking within a single document where missing some occurrences of morphological variants of search terms has an effect on window placement and system performance – further investigation is needed

The use of the *stddevselection* method for selecting the centre point of a passage typically produced better results then the *meanselection* method. That is, there are, indeed, outliers in the document that affect window placement.

4. Link-the-Wiki

In 2007 INEX introduced a new track, Link-the-Wiki. The aim is to automatically identify hypertext links for a new documents when added to a collection [3]. The task contains two parts, the identification of out-going links to other documents in the collection and the identification of in-going links from other documents to the new document. In keeping with the focused retrieval theme, links are from passages of text (anchor text) to best entry points in a target document. In 2007, as the task is new, a reduced version of the track was run in which the task is simply document to document linking (both incoming and outgoing) [3]. Participants were also asked to supply information about the specifications of the computer used to generate the results, and the time taken to perform the generation. We used Intel Pentium 4, 1.66GHz, single core, no hyper-threading, and only 512MB memory. Our execution times were all less than 4 minutes and are presented in Table 5.

4.1. Themes

Almost all words or phrase in a document could be linked to another document (if for no other reason than to define the term). The task, therefore, is not the identification of links, but the identification of salient links. The approach we took was the identification of themes (terms) that are over-represented within the document, and the identification of documents about those themes. Our approach is based on that of Shatkay & Wilbur [8].

An over-represented term is a term that occurs more frequently in the source document than expected, that is, the document is more about that term that would be expected if the term was used *ordinarily*. The actual frequency (*af*) of a term within the document is computed as the term frequency (*tf*) over the document length (*dl*).

$$af = \frac{tf}{dl}$$

The expected frequency (*ef*) of the term is computed on the prior assumption that the term does occur within the document. Given the collection frequency (*cf*) and the document frequency (*df*), and the average length of a document (*ml*), this is expressed as

$$ef = \frac{cf}{df \times ml}$$

The amount by which the term is over-represented (*repval*) in the document is the ratio of the actual frequency to the expected frequency.

$$repval = \frac{af}{ef}$$

Terms that occur in a document but not the collection are assigned negative scores.

4.2. Link-the-Wiki Runs

We generated document to document linking runs using a relevance ranking search engine that used BM25 (*k1*=0.421, *k3*=242.61, *b*=0.498). Incoming links and outgoing links were strictly reciprocal, that is, the list of incoming links was generated from the outgoing list by reversing the direction of each link (and maintaining the relative rank order).

First the source (orphan) document was parsed and a list of all unique terms and *repval* scores was generated. Stop words were removed from the list.

Five runs were generated from the term list. In the first the single most over-represented term was used to generate a query for which we searched the collection returning the top 50 documents. The second term was then used to identify the next 50 documents, and so on until 250 documents had been identified.

In the second run the top two terms were used and 100 documents identified. 100 more for the third and fourth term, and 50 for the sixth and seventh term. In the third run triplets of terms were used to identify 150 documents each. In the fourth run quads of terms were used, and in the final run sets of 5 terms were used to identify all 250 documents. The details are outlined in Table 5.

In our experiment the total length of the result set was held constant (at 250) and the number of documents retrieved per search terms was held constant (at 50). The aim of our experiment was to identify whether or not there was a query-length effect in identifying related documents.

Table 5. Runs submitted to the Link-the-Wiki track.

Run	Query length	Results per query	Time
ltw-one	1	50/50/50/50/50	134s
ltw-two	2	100/100/50	170s
ltw-three	3	150/100	161s
ltw-four	4	200/50	225s
ltw-five	5	250	124s

4.3. Results

The performance of the runs measured using mean average precision (MAP) is presented in Table 6. The relative rank order of our runs for both incoming and outgoing links was the same. The best run we submitted performed 4th of 13 submitted runs.

Figure 6 graphs outgoing precision (and Figure 7 incoming precision) at early points in the results list. Comparing the two, the technique we used is far better at identifying incoming links than outgoing links. When compared to runs from other participants, our best incoming precision at 5 and 10 documents ranked first.

Table 6: Link-the-Wiki results.

Run	Outgoing		Incoming	
	MAP	Rank	MAP	Rank
ltw-four	0.102	4	0.339	4
tw-five	0.101	5	0.319	5
ltw-three	0.092	7	0.318	6
ltw-two	0.081	8	0.284	7
ltw-one	0.048	13	0.123	9

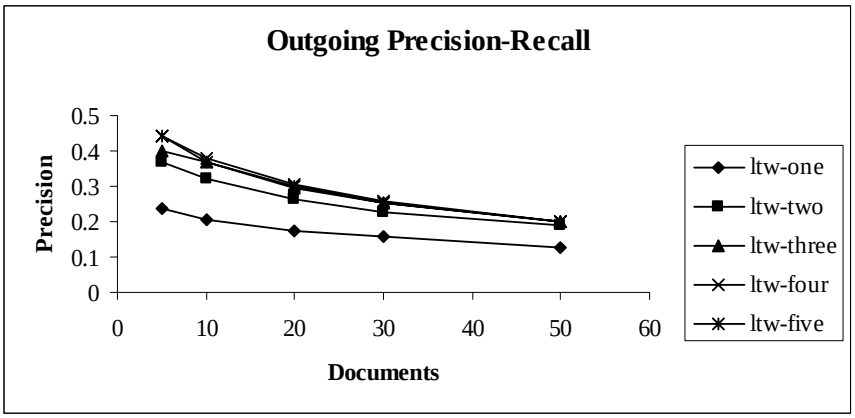

Figure 6: Precision – Recall of outgoing links.

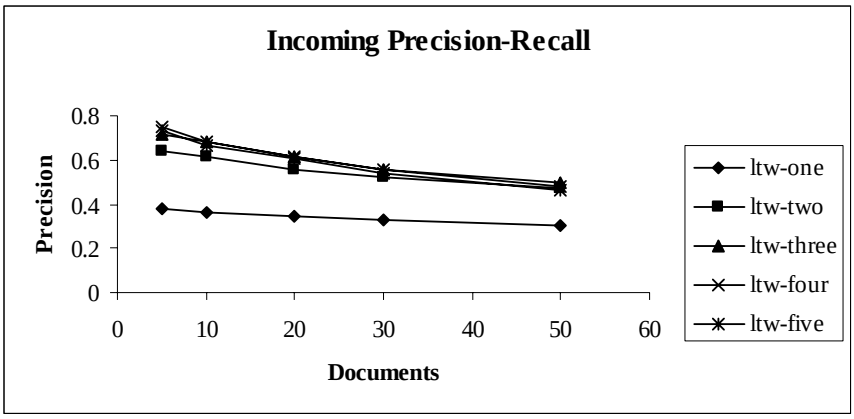

Figure 7: Precision – Recall of incoming links.

4.4. Discussion

We experimented with queries of different length and discovered that queries of 4 terms work better than either longer or shorter queries. When adding search terms to a query there comes a point at which the query becomes general resulting in the retrieval any an increasing number of irrelevant documents. This point appears to be 4 terms.

Of particular interest to us is the difference in performance of incoming and outgoing links. We constructed outgoing links from a document using a simple technique to identify terms that were over-represented. Incoming links were simply the same list inverted in direction. The technique appears capable of identifying the salient concepts within the document (such that it might be beneficial to link to), but not extracting from a document concepts that require further details (such that it might be beneficial to link from).

Our results suggests a future strategy in which the technique we used is applied to all documents to identify incoming links, and flipping those to get outgoing links for a document. This is, however, likely to be computationally expensive.

5. Conclusions

Passage retrieval and link discovery in the Wikipedia was examined in the context of INEX 2007. For both tasks methods that ignored document structure were studied. We found mixed results for both stemming and outlier reduction with no evidence that either was always effective. In link discovery we found that queries containing 4 search terms was effective.

In future work we intend to extend our methods to include document structures. Others have already shown that relevant passages typically start and end on tag boundaries, none the less we chose to ignore structure. Methods of using structure in

passage length identification will be examined for passage retrieval and use for Best Entry Point identification will be used for link identification.

We intent to examine the granularity of structural markup necessary before good ranking performance can be expected. Even though we chose to ignore structure the performance of our runs was reasonable when compared to those of others. This raises the question of the value of the structural markup within a document when used for relevance ranking.

The Link-the-Wiki runs we submitted also performed adequately. Queries of various length were constructed from concept terms. The concept terms were extracted from the orphaned document by taking terms overly represented in the document. The best query length we found was 4 terms.

The technique was better at identifying incoming links than outgoing links – that is, the technique identifies the concepts of the document and not concepts that require further expansion. Future work will examine fast and efficient ways to identify outgoing links.

6. Acknowledgements

Funded in part by a University of Otago Research Grant.

7. References

[1] Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor: University of Michigan Press.

[2] Huang, W., Trotman, A., & O'Keefe, R. A. (2006). Element retrieval using a passage retrieval approach. *Australian Journal of Intelligent Information Processing Systems (AJIIPS)*, 9(2):80-83.

[3] Huang, W. C., Trotman, A., & Geva, S. (2007). Collaborative knowledge management: Evaluation of automated link discovery in the Wikipedia. In *Proceedings of the SIGIR 2007 Workshop on Focused Retrieval*, 9-16.

[4] Kamps, J., & Koolen, M. (2007). On the relation between relevant passages and XML document structure. In *Proceedings of the SIGIR 2007 Workshop on Focused Retrieval*, 28-32.

[5] Kamps, J., Koolen, M., & Lalmas, M. (2007). Where to start reading a textual XML document? In *Proceedings of the 30th ACM SIGIR Conference on Information Retrieval*.

[6] Porter, M. (1980). An algorithm for suffix stripping. *Program*, 14(3):130-137.

[7] Robertson, S. E., Walker, S., Beaulieu, M. M., Gatford, M., & Payne, A. (1995). Okapi at TREC-4. In *Proceedings of the 4th Text REtrieval Conference (TREC-4)*, 73-96.

[8] Shatkay, H., & Wilbur, W. J. (2000). Finding themes in medline documents probabilistic similarity search. In *Proceedings of the Advances in Digital Libraries*, 183-192.

[9] Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted document length normalization. In *Proceedings of the 19th ACM SIGIR Conference on Information Retrieval*, 21-29.

[10] Tombros, A., Larsen, B., & Malik, S. (2004). The interactive track at INEX 2004. In *Proceedings of the INEX 2004 Workshop*, 410-423.

[11] Trotman, A., Geva, S., & Kamps, J. (2007). *Proceedings of the SIGIR 2007 workshop on focused retrieval*.

[12] Trotman, A., Pharo, N., & Jenkinson, D. (2007). Can we at least agree on something? In *Proceedings of the SIGIR 2007 Workshop on Focused Retrieval*, 49-56.

[13] Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *Transactions on Information Systems*, 22(2):179-214.