# Invited Talks

**Time-Biased Gain**
**Charles L. A. Clarke (University of Waterloo)**

Time-biased gain provides a unifying framework for information retrieval evaluation, generalizing many traditional effectiveness measures while accommodating aspects of user behaviour not captured by these measures. By using time as a basis for calibration against actual user data, time-biased gain can reflect aspects of the search process that directly impact user experience, including document length, near-duplicate documents, and summaries. Unlike traditional measures, which must be arbitrarily normalized for averaging purposes, time-biased gain is reported in meaningful units, such as the total number of relevant documents seen by the user. In work reported at SIGIR 2012, we proposed and validated a closed-form equation for estimating time-biased gain, explored its properties, and compared it to standard approaches. In work reported at CIKM 2012, we used stochastic simulation to numerically approximate time-biased gain, an approach that provides greater flexibility, allowing us to accommodate different types of user behaviour and increases the realism of the effectiveness measure. In work reported at HCIR 2012, we extended our stochastic simulation to model the variation between users. In this talk, I will provide an overview of time-biased gain, and outline our ongoing and future work, including extensions to evaluate query suggestion, diversity, and whole-page relevance. This is joint work with Mark Smucker.

**The Challenges of Building Online Community Museums**
**Nigel Stanger (University of Otago)**

Over the past few years I have been involved in the development of online museums for two different communities in Central Otago, New Zealand. Developing digital archives for the general public raises some interesting issues, including usability, copyright, curation of items, and potentially dealing with unusual document types. In this talk I will discuss these and other issues, and chronicle the development of both museums.

## Symposium Chair

Andrew Trotman, University of Otago

## Programme Co-chairs

Sally Jo Cunningham, University of Waikato
Laurianne Sitbon, Queensland University of Technology

## Program Committee

Robert Allen, Victoria University of Wellington
Peter Bruza, Queensland University of Technology
Shane Culpepper, RMIT University
Sally Jo Cunningham, University of Waikato
David Eyers, University of Otago
Shlomo Geva, Queensland University of Technology
David Hawking, Funnelback
Yun Sing Koh, University of Auckland
Irena Koprinska, University of Sydney
Alexander Krumpholz, CSIRO/ ANU
Alistair Moffat, University of Melbourne
Glen Newton, Carleton University
Laz Park, University of Western Sydney
Luiz Augusto Pizzato, University of Sydney
Gitesh Raikundalia, Victoria University
Mark Sanderson, RMIT University
Falk Scholer, RMIT University
Laurianne Sitbon, Queensland University of Technology
Ling-Xiang Tang, Queensland University of Technology
James A. Thom, RMIT University
Andrew Trotman, University of Otago
Andrew Turpin, University of Melbourne
Mingfang Wu, RMIT University
Burkhard Wuensche, University of Auckland
Guido Zuccon, CSIRO

## ADCS Steering Committee

Sally Jo Cunningham, University of Waikato
Shlomo Geva, Queensland University of Technology
Mark Sanderson, RMIT University
Falk Scholer, RMIT University
James A. Thom , RMIT University
Paul Thomas, CSIRO
Andrew Trotman , University of Otago
Andrew Turpin , University of Melbourne

# Proceedings of the Seventeenth Australasian Document Computing Symposium
# University of Otago, Dunedin, New Zealand
# 5-6 December 2012

## Chair's Preface

These proceedings contain the papers of the Seventeenth Australasian Document Computing Symposium hosted by the University of Otago and held in Dunedin, New Zealand.

The quality of submissions was again very high this year. Of the 24 papers submitted, 11 were accepted for full presentation at the symposium (46%) and 8 were accepted for short presentation (32%). The full written version of each submission received three anonymous reviews by independent, qualified international experts in the area. Dual submissions were explicitly prohibited.

We would like to thank the members of the program committee for their reviewing efforts. We would also like to thank ACM SIGIR, Google, NICTA, Bing, Funnelback, and the University of Otago for their generous support of the event.

The symposium includes many formal presentations, but perhaps its greatest benefit lies in the opportunity it provides for document computing practitioners and researchers to get together and informally share ideas. Once again we have collocated with The Australasian Language Technology Workshop (ALTA), sharing a joint paper session, a poster session, and social events.

Andrew Trotman (Chair)
7 November 2012

# Table of Contents

## Full Papers



## Short Papers

# Effects of Spam Removal on Search Engine Efficiency and Effectiveness

Matt Crane
Department of Computer Science
University of Otago
Dunedin, New Zealand
mcrane@cs.otago.ac.nz

Andrew Trotman
Department of Computer Science
University of Otago
Dunedin, New Zealand
andrew@cs.otago.ac.nz

## ABSTRACT

Spam has long been identified as a problem that web search engines are required to deal with. Large collection sizes are also an increasing issue for institutions that do not have the necessary resources to process them in their entirety. In this paper we investigate the effect that withholding documents identified as spam has on the resources required to process large collections. We also investigate the resulting search effectiveness and efficiency when different amounts of spam are withheld. We find that by removing spam at indexing time we are able to decrease the index size without affecting the indexing throughput, and are able to improve search precision for some thresholds.

## Categories and Subject Descriptors

H.3.1 [**Information Search and Retrieval**]: Content Analysis and Indexing – Indexing methods; H.3.3 [**Information Search and Retrieval**]: Information Filtering

## Keywords

Information Retrieval, Web Documents, Spam, Procrastination

## 1. INTRODUCTION

Spam has been a long identified problem that web search engines must address. In 2009 TREC adopted the ClueWeb09 collection, a crawl of 1 billion web pages, as a standard collection for web track tasks. Some TREC submissions also made use of proprietary spam filters in their submissions [11].

Zuccon *et al.* [16] investigated the effect of withholding documents identified as spam on indexing and retrieval performance. They presented some interesting results, including a $u$-shaped relationship between amount of spam withheld from the index and the indexing time. They also show that there is no effect on retrieval time when spam was removed.

We aim to reproduce and explain these results, and to extend them by performing our own experiments of the effectiveness and efficiency that withholding spam documents has on a search engine.

We find that we are unable to replicate some of their results, but present some plausible reasons for them. Specifically, we find that the indexing time decreases consistently when more documents are excluded, and that retrieval time is strongly correlated with the size of the index that is generated.


*ADCS'12,* December 5-6, 2012, Dunedin, New Zealand


## 2. RELATED WORK

Cormack *et al.* [8] provide an in-depth examination of the effects of spam on the performance of the runs submitted to TREC 2009. They generated four different rankings of the spamminess of pages within the English ClueWeb09 dataset:

- **UK2006:** A set of labels trained against a small set of web pages containing 746 spam pages and 7,474 non-spam pages.
- **Britney:** Derived from results returned for popular queries given to commercial search engines.
- **Group X:** Manually labelled from results for queries from the 2009 TREC Ad-hoc task.
- **Fusion:** A combination of the other three methods.

Using these filters Cormack *et al.* were able to improve almost all runs that were submitted to TREC 2009. Two methods for modifying the submitted runs were consequently proposed, both of which were seen to improve performance. The first was to discard documents from the result set that did not meet a minimum threshold for non-spamminess, the second was to re-rank results using the spam score as a feature of the document.

The spam scores generated by Cormack *et al.* have subsequently been made available for use by other researchers, and have subsequently been used in a number of the top ranking systems in both the TREC 2010 and 2011, Web tracks for both the Ad-hoc and Diversity tasks as a threshold for indexing [1, 10, 13], a threshold for post-processing of returned results [9], and a feature to be used in document ranking [2, 12].

In 2010 TREC ran a spam identification track [5], for which the Fusion ranking generated by Cormack *et al.* was used a baseline. This baseline was not bettered.

| Year | Number of Queries | TREC Query Numbers | Mean Query Length |
|---|---|---|---|
| 2009 | 50 | 1–50 | 2.1 |
| 2010 | 48 | 51-99 | 2.02 |
| 2011 | 50 | 101–150 | 3.4 |

**Table 1: Statistics for query sets being used.**

| Collection | ClueWeb09 Category B |
|---|---|
| Documents | 50,220,423 |
| Size | 1.5TB |
| Unique Terms | 96,298,556 |
| Total Terms | 75,614,656,698 |
| Mean Document Length | 1,505.66 |

**Table 2: Collection statistics for ClueWeb09 Category B.**

Zuccon *et al.* [16] investigated the effect that removal of spam had on the resources required to index ClueWeb09 Category B — the first 50 million English documents — and the effectiveness of the resulting indexes.

As Zuccon *et al.* stands, to our knowledge, as the only systematic investigation of the effects that spam removal at indexing time has on indexing performance and subsequent retrieval performance we seek to replicate and extend their experiments with the aim of applying the results learned on Category B to the Category A collection.

Intuitively removing documents from the indexing process will result in both a smaller index, and less time required to index. This was the opposite of the results presented in Zuccon *et al.*, who saw an increase in indexing time when removing a large proportion of the collection. Removing documents from the indexing process will also have an effect on the retrieval performance of the system, however, Zuccon *et al.* found no change in the retrieval time for a selection of ranking functions.

For our experiments we use the ATIRE search engine [14][1]. The experiments conducted in this paper are all performed on a machine with a quad cpu AMD Opteron 6276 2.3GHz 16-core, 512GB PC12800 memory, 6× 600GB 10000 RPM hard drives, and running Linux with kernel version 2.6.32.

Table 2 shows statistics of ClueWeb09 Category B. Table 1 shows some of the statistics for the query sets, we excluded queries 95 and 100 from the 2010 query set because no relevance judgements are available for them.

## 3. INDEXING

Zuccon *et al.* [16] propose an algorithm for modifying the indexing process to consider the spam score of the document being indexed, and only index those documents where its score met a given threshold. A list of documents to exclude is constructed prior to indexing, and this is consulted to determine whether to index a document.

Their results show that the indexing time forms a $u$-shape with respect to the given threshold, indicating that a higher threshold would take longer to index than a lower threshold.

One of the driving design decisions behind the ATIRE search engine [14] is the absence of any preprocessing. To keep aligned with this design decision, the spam filtering we added to ATIRE generates an internal list of documents to exclude at run-time from the complete list of <document, score> pairs. We, like Zuccon *et al.*, use the Cormack *et al.*'s Fusion scores.

However, we additionally take advantage of the fact that the spam scores are percentile scores. This allows us to construction an inclusion list if the threshold is greater than 50.

For instance, if given a threshold of 70, on the Category A collection 352,732,667 documents would be excluded from the index. Doing a binary search on this list of docids would require 29 string comparisons to identify whether a document should be excluded, compared with 27 when searching the list of documents that should be included. These comparisons are done on every document that is encountered during indexing, saving a total of over 1 billion string comparisons when indexing Category A.

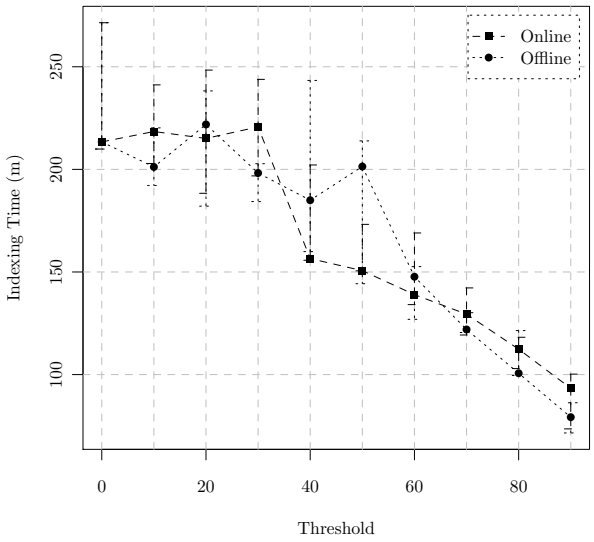



**Figure 1: Comparative indexing time between online and offline calculation of documents to include or exclude from indexing with respect to the spam threshold given.**

Figure 1 shows the total indexing time for different thresholds using both the online and offline generation of lists. The figure shows the results across three runs for each method, with the median runs being joined and the slower and faster runs shown as error bars.

In an attempt to replicate the results from Zuccon *et al.* the offline method calculates only lists of documents to exclude. However, instead of the $u$-shaped results, we instead see a consistent drop-off in indexing times using both methods.

The reasons for the $u$-shape as seen in Zuccon *et al.* [16] is unclear and we were unable to reproduce it. They suggest that it "... may be caused by the procedure we used for loading the file containing the list of documents which do not have to be considered ...".

[1]Changeset: 56591fcef100

| Threshold | Documents | Unique Terms | Total Terms | Mean Document Length |
|---|---|---|---|---|
| 0 | 50,220,423 | 96,298,556 | 75,614,656,698 | 1,505.66 |
| 10 | 48,736,112 | 74,805,408 | 72,664,843,957 | 1,490.99 |
| 20 | 46,432,700 | 69,693,504 | 69,021,947,305 | 1,486.49 |
| 30 | 43,718,178 | 64,173,727 | 64,714,615,144 | 1,480.27 |
| 40 | 40,844,719 | 58,008,256 | 60,003,421,198 | 1,469.06 |
| 50 | 37,655,996 | 51,750,917 | 54,792,100,355 | 1,455.07 |
| 60 | 33,836,981 | 45,082,561 | 48,802,927,422 | 1,442.30 |
| 70 | 29,038,220 | 37,621,793 | 41,525,104,236 | 1,430.02 |
| 80 | 23,148,047 | 29,487,702 | 32,832,607,822 | 1,418.37 |
| 90 | 15,374,591 | 19,745,587 | 21,417,223,927 | 1,393.03 |

**Table 3: Index statistics for indexes generated with different threshold values.**

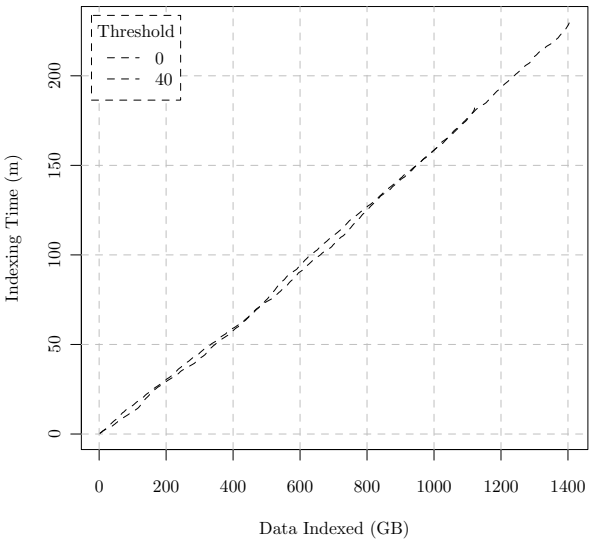



**Figure 2: Indexing throughput for two different thresholds of spam removal as measured by time taken relative to data indexed.**

We believe this is a reasonable explanation as Zuccon *et al.* use C++'s `>>` operator to read documents into a `std::map`, while the spam filter added to ATIRE performs a block read of the file, and performs a linear scan to set up pointers with no unnecessary copying of data.

We also make note of the large difference in total indexing times (≈1750 minutes compared with 210) between the two indexing processes when performing no spam filtering. This suggests that there may also be an underlying engineering component to the discrepancy in results.

We see only a marginal overhead in the online calculation of the lists of docids to discard or keep, with the difference between median runs with a threshold of 70 being 7 minutes, or ≈6%.

Figure 2 shows the time taken to index as a function of the amount of data that has been indexed (without spam removal, and with a threshold of 40). This figure shows that by withholding spam from the index, we do not substantially affect the throughput rate of the indexing system, which indicates that the time taken to determine whether to index a document is negligible when compared to the time needed to index that document. Further evidence that the $u$-shape is due to factors outside of the spam identification itself.

We also note from this graph the near linear relationship between indexing time and data indexed ($r^2$ value of 0.9985).

Figure 3 shows the resulting index sizes for each of the indexes generated. We see a similar drop off in relative index sizes as Zuccon *et al.* [16], with a threshold of 40 generating an index that is 24GB in size, while their index is ≈135GB for a threshold of 45.

Table 3 show some statistics — number of documents, number of unique terms, number of total terms, and average document length — of the generated indexes for each different threshold, for example at a threshold of 40 we index 41 million documents, containing 60 billion instances of 58 million unique terms and an average document length of 1,469 terms.

Interestingly, we see a sharper drop off in unique terms than total terms, as well as a consistent decline in the average document length. This indicates that documents that are identified as spam have a higher proportion of unique terms, and tend to be longer.

The ATIRE search engine defaults to Variable Byte compression. The ATIRE search engine also stores the postings lists using impact ordering on term frequency, which is itself a form of compression [14].

We note that the number of documents remaining in the index does not match the thresholds given. This is because the spam scores were generated across the complete set of 500 million English documents of ClueWeb09 Category A. Category B contains the first documents crawled, and due to the nature of the crawl contains documents that are less likely to be spam. If one wanted to remove half the documents in Category B, then a threshold in the 70s should be specified.

# 4. SEARCHING

## 4.1 Effectiveness

Having now investigated the performance of indexing under different thresholds for spam removal, we now investigate the efficacy of the search engine across these different indexes. We measure search performance against the qrels from the diversity task for the 2009 queries to enable training on these queries and testing on the 2010 and 2011 queries.

As a precision measure we use ERR-IA as described by Chapelle *et al.* [4] as the primary measure as it is used by TREC. We also report $\alpha$-nDCG [7] scores to enable compar-

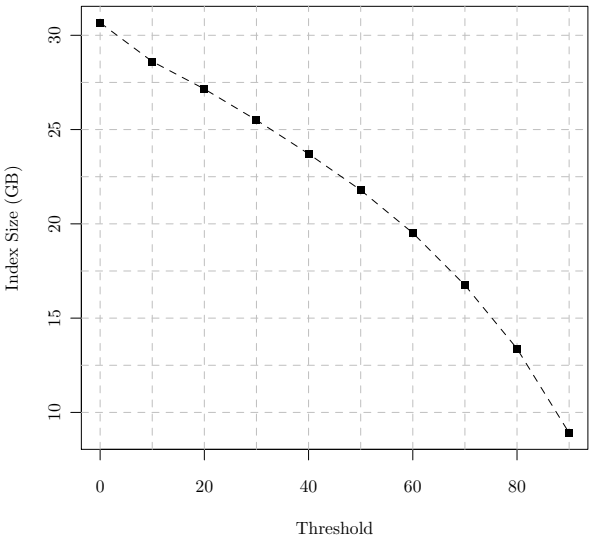



**Figure 3: Size of generated index for different threshold values.**

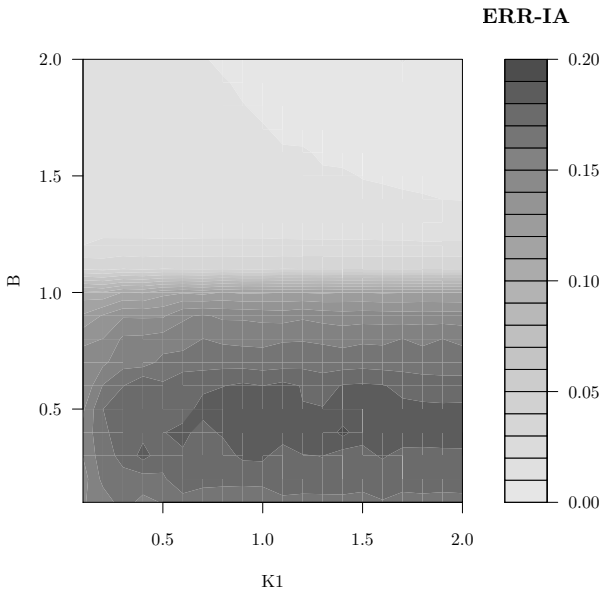



**Figure 4: Example surface generated from the grid search of BM25 parameters on index with spam threshold set to 40. A darker shade indicates a higher ERR-IA@20.**

ison with prior reported results.

For each index we perform a grid search to find the best parameters for the BM25 function. The grid search is performed across the 50 queries from the 2009 web track at TREC, with ERR-IA@20 as the primary performance measure.

Figure 4 shows an example surface generated by the grid search performed on the index with spam threshold set to 40. The darker the shading, the higher the ERR-IA@20, with a peak value of 0.1931 when k1=1.4 and b=0.4. The other, traditional, variables within the BM25 ranking functions are ignored due to the modified version implemented within the ATIRE search engine [14].

Zuccon *et al.* [16] identified a threshold of 70 provided the best results when considering both ERR-IA@10 and $\alpha$-nDCG@10 across a range of ranking functions. Cormack *et al.* [8] identified a threshold of 50 for Category B when altering runs submitted to TREC. Both identify an upside down $u$-shape, with no filtering performing the worst.

Figure 5 shows the results from the grid searches as a function of the spam threshold specified during indexing. These are optimal scores with our ranking function. We identify the same upside down $u$-shape, although we find that the performance of no spam filtering to perform better than a threshold of 90. When targeting ERR-IA@20 we obtain a peak value of 0.1931 when a threshold of 40 is given at indexing time.

Figure 6 shows the results of evaluating using $\alpha$-nDCG@20 when using the BM25 parameters found from the grid search against ERR-IA@20. We find the same upside down $u$-shape, suggesting that this is evaluation function-independent. Interestingly, we find that no spam filtering performs better than specifying a threshold of 70 or higher. A peak $\alpha$-nDCG@20 value of 0.3237 is obtained with a threshold of 30. These are optimal scores using our ranking

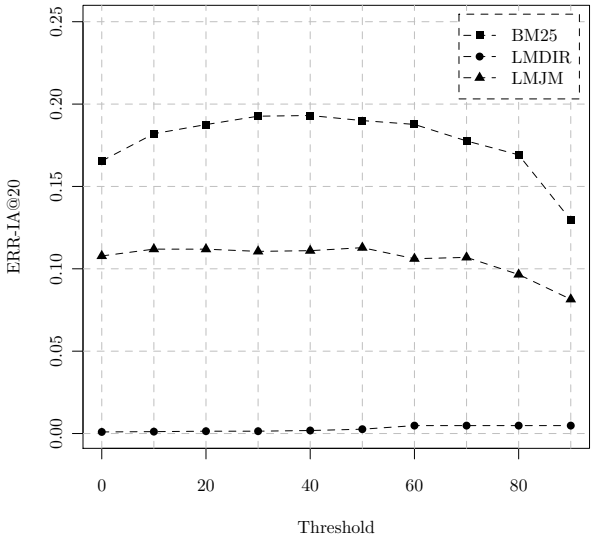



**Figure 5: Best results from grid search of BM25 values for different thresholds targeting ERR-IA@20 as the evaluation function.**

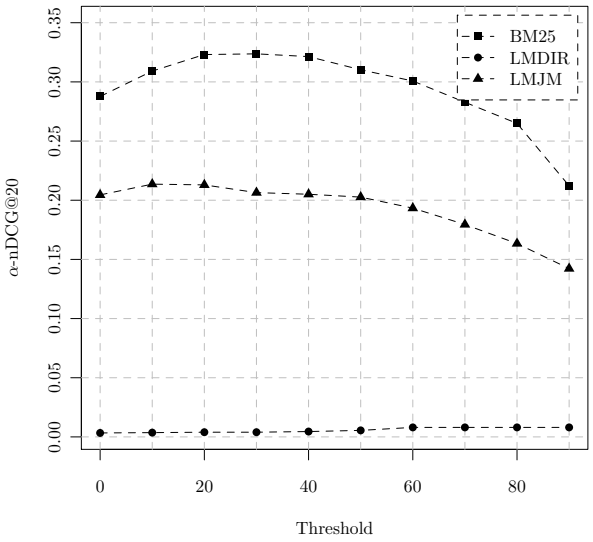



**Figure 6: Best results from grid search of BM25 values for different thresholds using $\alpha$-nDCG@20 as the evaluation function from search against ERR-IA@20.**

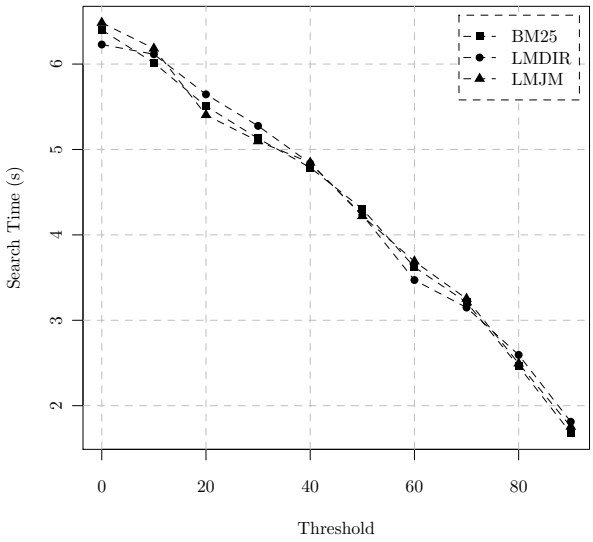



**Figure 7: Average time taken to search with respect to spam removal threshold.**

function.

Because of its use as the primary ranking function in the TREC diversity tasks, the rest of the paper focuses on ERR-IA@20, but will reports $\alpha$-nDCG values as appropriate.

## 4.2 Efficiency

Zuccon *et al.* [16] identify an interesting phenomenon when evaluating the time to search, where only the LMJM [15] ranking function (Unigram Language Model with Jelineker-Mercer smoothing) showed any change in query throughput. They suggest that this change in throughput is caused by implementations in Indri. Intuitively, however, a smaller index *should* result in higher throughput, with time taken to search 0 documents being near 0.

Figure 7 shows the retrieval time per query for indexes pruned of spam (at various thresholds). We see an almost linear drop off in relation to the spam threshold regardless of ranking function. For instance removing no spam results in an average search time of 6 seconds, while a threshold of 90 results in an average search being performed in a quarter of the time.

Times for BM25 were averaged across 400 runs of the 50 queries used during the grid search, while LMJM and LMDIR [15] (Unigram Language Model with Dirichlet smoothing) times were averaged across 3 runs of the same queries. For LMJM and LMDIR, we utilised the same parameters as Zuccon *et al.*, that is, $\mu = 3000$ for LMDIR, and $\lambda = 0.01$ for LMJM.

We cannot suggest a reason for no change in retrieval performance identified by Zuccon *et al.*, other than their own, specific implementations in Indri.

# 5. FURTHER REDUCTIONS

Having generated an index that provides good performance on the 2009 queries, we now question whether we can further reduce the size of the index without compromising on the precision of the results.

We can further reduce the size of the index by stopping terms. The selection of these terms can influence the performance of the search engine, by removing potential query terms, and by altering the statistics of the documents that are indexed. We select numbers and HTML tags as obvious candidates for removal.

By stopping numbers we are able to reduce the size of the index by 3GB, by stopping tags we can reduce the size of the index by 1GB. Stopping numbers reduces ERR-IA@20 to 0.1827. Whereas stopping the tags has no effect on retrieval performance, because ATIRE does not consider the presence of tags in the document or collection statistics.

The terms that are of the most importance when considering a document's relevance to a query are the "middle" terms, that is, those terms that are not infrequent, and also not frequent. To this end, we stop those terms with a document frequency of 1, reducing the index size by 1GB and resulting in an ERR-IA@20 of 0.1931.

When all of the stopping conditions are selected, the resulting index is reduced in size by 20%. This is slightly less than the sum of the individual improvements because some terms may be stopped under multiple conditions. This reduction in index size is accompanied by an approximate 15% decrease in time required to search.

Unfortunately, the ERR-IA@20 for the stopped index drops from 0.1931 to 0.1826, which is a statistically significant change ($p$-value $< 0.01$ on a two-tailed pairwise $t$-test). We consider this change in performance to be acceptable given the increase in efficiency. This efficiency versus effectiveness trade-off has been explored by others [3].

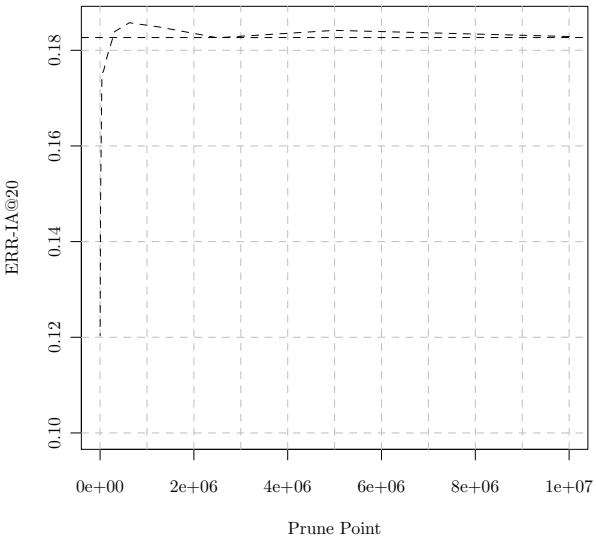



**Figure 8: Effect of static pruning on ERR-IA@20 on index with numbers, tags and infrequent terms removed and spam threshold set to 40.**

| | ERR-IA | | $\alpha$-nDCG | |
|---|---|---|---|---|
| Year | @10 | @20 | @10 | @ 20 |
| 2009 | 0.1792 | 0.1842 | 0.2892 | 0.3105 |
| 2010 | 0.1863 | 0.1995 | 0.2533 | 0.3029 |
| 2011 | 0.4035 | 0.4123 | 0.4803 | 0.5120 |

**Table 4: Evaluation scores on the different query sets using BM25 learned on 2009 queries, with spam threshold set to 40.**

We can further reduce the size of the index by statically pruning the postings lists. That is, only allowing at most the first $n$ documents of each postings list for a term to remain in the index. Recall that our index is impact ordered on term frequency and so the first documents have the highest tf scores. The ATIRE search engine support this at both search-time and index-time parameters, allowing us to train at search time and to re-index with the optimal value.

Figure 8 shows the ERR-IA@20 as a function of this pruning value, with the stopped, unpruned performance shown as the black dashed line. We see that it takes substantial pruning before the retrieval performance is degraded, and some pruning actually increases the performance. A pruning value of 300,000 provides an ERR-IA@20 of 0.1840 with a $p$-value of 0.051 (not significant) when compared to the unstopped, unpruned index.

When pushing this static pruning to indexing time, we reduce the size of the index by only 400MB. This relatively small change in the index size is accompanied by a much larger relative reduction of search time from 5 seconds, to 850 milliseconds, when our index is stored on disk. This is because the postings lists do not take much space inside the index, but are processed exhaustively during search.

The results for all query sets on this final generated index

| | ERR-IA | | $\alpha$-nDCG | |
|---|---|---|---|---|
| Year | @10 | @20 | @10 | @ 20 |
| 2009 | 0.1044 | 0.1105 | 0.1732 | 0.2021 |
| 2010 | 0.1048 | 0.1124 | 0.1702 | 0.1968 |
| 2011 | 0.2934 | 0.2995 | 0.3667 | 0.3883 |

**Table 5: Evaluation scores on the different query sets using LMJM, $\lambda = 0.01$, with spam threshold set to 40.**

| | ERR-IA | | $\alpha$-nDCG | |
|---|---|---|---|---|
| Year | @10 | @20 | @10 | @ 20 |
| 2009 | 0.0026 | 0.0026 | 0.0056 | 0.0055 |
| 2010 | 0.0004 | 0.0014 | 0.0017 | 0.0060 |
| 2011 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

**Table 6: Evaluation scores on the different query sets using LMDIR, $\mu = 3000$, with spam threshold set to 40.**

are shown in Table 4. The scores for 2009 should not be used in direct comparison as these were the queries used for training, they are included for completeness. On the 2009 queries, the final index has a $p$-value of 0.04 when compared to the index generated with no filtering applied.

We note that the results on the 2011 queries would have placed as $6^{th}$ in the diversity task at TREC [6] without performing any explicit diversification and using just BM25 for ranking. We do, however, concede that this is not equivalent to submitting a run to TREC.

Table 7 shows the results of applying the same spam thresholding, stop word removal and static pruning on the Category A collection. The results for a threshold of 70 are also shown, as this has been shown in previous work to provide the best threshold for this collection [8]. The parameters for BM25 were again trained on the 2009 query set by grid search against ERR-IA@20. The thresholds of 40 & 70 yield index sizes of 120GB and 62GB and allow searching in an average time of 4.5 and 1.8 seconds respectively.

# 6. FUTURE WORK

We can further reduce the overhead of the search to find if a document should be excluded at indexing time. The files within each of the `.warc.gz` files are stored in sequential docid order. By performing a binary search when we inspect the first document within the archive, and then using a linear scan for the rest of the archive we hypothesise that we can drastically reduce the number of string comparisons required.

The index size can also be further decreased by utilising stemming at indexing time. This would have the effect of conflating terms together and reducing the size of the dictionary. However, this would require re-learning BM25 weights, a good static pruning value, and which terms could still be effectively stopped. The inclusion of more items to consider at indexing time quickly turns this process into a multi-objective optimisation problem.

In exploring the effect of the spam score on search effectiveness, Cormack *et al.* [8] found that by re-ranking results by incorporating the spamminess score improved precision. Indeed this approach has been taken by a number of top ranking runs in the TREC 2011 web track tasks [2, 12]. We

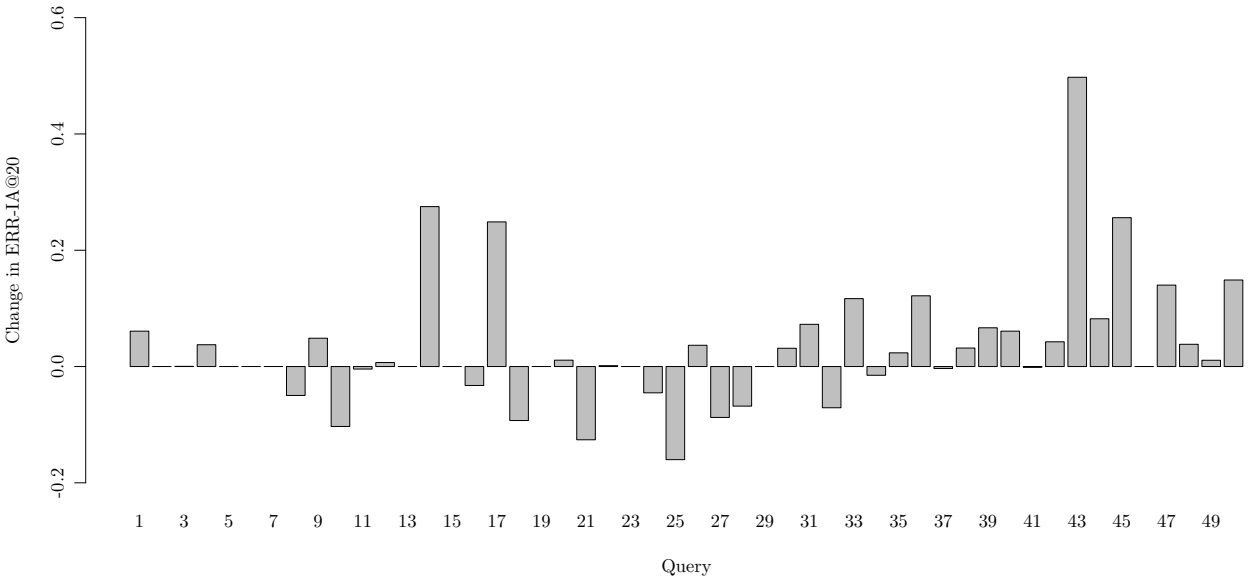



**Figure 9: Change in ERR-IA@20 for individual queries in the 2009 query set from no filtering to final index. A positive value means an improvement for that query, while a negative value indicates performance degradation.**

| | | ERR-IA | | $\alpha$-nDCG | |
|---|---|---|---|---|---|
| Year | Threshold | @10 | @20 | @10 | @ 20 |
| 2009 | 40 | 0.1096 | 0.1178 | 0.1781 | 0.2069 |
| | 70 | 0.1217 | 0.1289 | 0.1902 | 0.2204 |
| 2010 | 40 | 0.1430 | 0.1509 | 0.1991 | 0.2289 |
| | 70 | 0.1778 | 0.1883 | 0.2384 | 0.2794 |
| 2011 | 40 | 0.3595 | 0.3718 | 0.4446 | 0.4878 |
| | 70 | 0.4067 | 0.4173 | 0.4846 | 0.5216 |

**Table 7: Evaluation scores on Category A between threshold learned on Category B with previous research, with learned stop word removal and static pruning.**

will explore this alternative approach to the spam problem.

The imminent release of ClueWeb12 provides an opportunity to examine the effects that spam removal will have on that corpus.

# 7. DISCUSSION & CONCLUSION

We have investigated the effect of removing spam documents on both indexing and retrieval performance. We found a consistent decrease in time to index as the amount of spam removed during indexing increased, and this was accompanied by a consistent decrease in index size. We also showed that by removing spam from the indexing process we did not alter the near linearity of the indexing system.

For each generated index, we then tuned our implementation of BM25 and identified the threshold that provided the best effectiveness when measuring ERR-IA@20 on the TREC 2009 web track queries. We then investigated the efficiency of searching these indexes, and found a clear relationship between index size and search throughput.

These results are in contrast to those found within Zuccon *et al.* [16], which suggests their observed behaviour was extra to the process of indexing and searching (e.g. reading the spam list).

We then further reduced the index size without decreasing effectiveness by introducing stopping and static pruning. We found that while stopping numbers, tags and terms with a document frequency of one made a statistically significant decrease in effectiveness, static pruning at 300,000 improved performance to a statistically insignificant level when compared to the unstopped, unpruned index.

The final index generated for Category B is 20GB, and allows searching to be carried out in 400 milliseconds when the index is loaded entirely into memory, averaged across 3 runs of the 2009 TREC queries.

The retrieval effectiveness of the final index is comparable to the top-ranking systems for the diversity task at TREC 2010 and 2011 despite only using BM25 without anchor text or PageRank scores.

Figure 9 shows the change in ERR-IA@20 for individual queries in the 2009 query set from performing no filtering to the final index as discussed in this paper. A positive value indicates that performance for that query improved, while a negative value indicates that the performance for that query was degraded.

We see that there are a few queries that are improved substantially, most notably query 43 — "the secret garden" — which was improved substantially. We also see a small reduction in the performance of a selection of queries that at cursory glance would suggest should be improved by spam filtering — 10: "cheap internet", 21: "volvo", 32: "website design hosting".

# 8. REFERENCES


[1] M. Bendersky, D. Fisher, and W. Croft. Umass at trec 2010 web track: Term dependence, spam filtering and quality bias. In *Proceedings of the Text REtrieval Conference (TREC 2010)*, 2010.

[2] B. Billerbeck, N. Craswell, D. Fetterly, and M. Najork. Microsoft Research at TREC 2011 Web Track. In *Proceedings of the Text REtrieval Conference (TREC 2011)*, 2011.

[3] S. Büttcher and C. Clarke. Efficiency vs. effectiveness in terabyte-scale information retrieval. In *Proceedings of the Text REtrieval Conference (TREC 2005)*, 2005.

[4] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 621–630. ACM, 2009.

[5] C. Clarke, N. Craswell, I. Soboroff, and G. Cormack. Overview of the TREC 2010 Web track. In *Proceedings of the Text REtrieval Conference (TREC 2010)*, 2010.

[6] C. Clarke, N. Craswell, I. Soboroff, and E. Vorhees. Overview of the TREC 2011 Web track. In *Proceedings of the Text REtrieval Conference (TREC 2011)*, 2011.

[7] C. Clarke, M. Kolla, G. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666. ACM, 2008.

[8] G. Cormack, M. Smucker, and C. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information retrieval*, 14(5):441–465, 2011.

[9] C. Hauff and D. Hiemstra. University of Twente @ TREC 2009: Indexing half a billion web pages. In *Proceedings of the Text REtrieval Conference (TREC 2009)*, 2009.

[10] J. Kamps, R. Kaptein, and M. Koolen. Using anchor text, spam filtering and wikipedia for web search and entity ranking. In *Proceedings of the Text REtrieval Conference (TREC 2010)*, 2010.

[11] J. Lin, D. Metzler, T. Elsayed, and L. Wang. Of Ivory and Smurfs: Loxodontan MapReduce experiments for web search. 2009.

[12] R. McCreadie, C. Macdonald, R. Santos, and I. Ounis. University of Glasgow at TREC 2011: Experiments with Terrier in Crowdsourcing, Microblog, and Web Tracks. In *Proceedings of the Text REtrieval Conference (TREC 2011)*, 2011.

[13] M. Smucker. Crowdsourcing with a crowd of one and other TREC 2011 crowdsourcing and web track experiments. In *Proceedings of the Text REtrieval Conference (TREC 2011)*, 2012.

[14] A. Trotman, X. Jia, and M. Crane. Towards an efficient and effective search engine. In *Proceedings of the SIGIR 2012 Workshop on Open Source Information Retrieval*, pages 40–47, 2012.

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

[16] G. Zuccon, A. Nguyen, T. Leelanupab, and L. Azzopardi. Indexing without spam. In *Proceedings of the 16th Australasian Document Computing Symposium (ADCS 2011)*, pages 6–13, 2011.

# An English-Translated Parallel Corpus for the CJK Wikipedia Collections

Ling-Xiang Tang
Queensland University of Technology
Brisbane, Australia
l4.tang@qut.edu.au

Shlomo Geva
Queensland University of Technology
Brisbane, Australia
s.geva@qut.edu.au

Andrew Trotman
University of Otago
Dunedin, New Zealand
andrew@cs.otago.ac.nz

## ABSTRACT
In this paper, we describe a machine-translated parallel English corpus for the NTCIR Chinese, Japanese and Korean (CJK) Wikipedia collections. This document collection is named *CJK2E Wikipedia XML corpus*. The corpus could be used by the information retrieval research community and knowledge sharing in Wikipedia in many ways; for example, this corpus could be used for experimentations in cross-lingual information retrieval, cross-lingual link discovery, or omni-lingual information retrieval research. Furthermore, the translated CJK articles could be used to further expand the current coverage of the English Wikipedia.

## Categories and Subject Descriptors
H.3.7 [**Information Storage and Retrieval**]: Digital Libraries – *collection.*

## General Terms
Documentation, Experimentation, Languages

## Keywords
Wikipedia, Corpus, English, Chinese, Japanese, Korean, cross-lingual information retrieval, cross-lingual link discovery, machine learning.

## 1. INTRODUCTION
Wikipedia is currently the largest freely available online multilingual encyclopaedia. It contains a large number of articles covering millions of topics and has articles in most written languages. However the different language versions of Wikipedia have evolved at different rates and are unbalanced in coverage (and sometimes differently biased in content). Among all the language versions, English Wikipedia is the largest with over 6,550,000 articles[1].

But an article may not be written in a user's preferred language, or the user may be looking for richer content than is available in their preferred language. In these cases our user may be able to, and prepared to, read in a second or subsequent language – if they could find the content.

To address the problem of finding content in multiple languages NTCIR launched CrossLink, the cross-lingual link discovery (CLLD) [1] track. The aim of this track at NTCIR-9 was to build a system that could automatically recommend hypertext links from English documents to relevant documents in Chinese, Japanese, and Korean. Such a system must not only recommend topically relevant documents to a source document, but must also suggest appropriate anchors.

Good approaches to CLLD were seen at NTCIR-9 Crosslink. We observed that most systems seen there used translation in some ways. Typically seen were: direct machine translation (of, for example, entities) or triangulation (for example, following links from an English source article to an English target article then finding the Chinese equivalent article thorough "language links" or entity translation) [1-6].

To lower the barrier of entry to research such as CLLD and other cross lingual Information Retrieval problems we created and present here a machine-translated parallel English Wikipedia corpus derived from the Chinese, Japanese, and Korean (CJK) Wikipedia collections currently being used at NTCIR-10 CrossLink-2. This new corpus was built by translating the CJK Wikipedia articles into English using an online machine translation service (specifically, Google Translate[2]).

This machine translated corpus could be used for many purposes, including (but not limited to): cross-lingual link discovery (CLLD); cross-lingual information retrieval; omni-lingual information retrieval; as well as machine learning; cross-lingual document categorisation and clustering; and machine translation itself.

Better, the translated CJK articles could be used to further expand the coverage of the English Wikipedia. Topics not covered in English could be added and topics inadequately covered in English could be expanded (albeit not automatically).

The remainder of this article describes our translated corpus *CJK2E Wikipedia XML corpus* (Version 1.0). Collection statistics, corpus creation procedures, and an experiment demonstrating its utility, are included.

[1]The article number was collected from the Wikipedia database dump taken on 4th January 2012.



*ADCS '12,* December 05 - 06 2012, Dunedin, New Zealand
Copyright 2012 ACM 978-1-4503-1411-4/12/12 … $15.00.

[2] http://research.google.com/university/translate/index.html

## 2. THE CJK (SOURCE) COLLECTIONS

### 2.1 Corpora Statistics

The CJK Wikipedia collections[3] we used as the source for translation are those used in the NTCIR-10 Crosslink-2 track[4]. The collections were created from the Wikipedia XML dumps taken in January 2012. The original article text with Wikipedia mark-up was converted to XML using the YAWN system [7]. The details of the source collections are given in Table 1. The first column lists the language, the second column lists the number of documents in the dump, the third gives the size of the collection and the fourth column lists the dump date. For example, the Chinese dump taken on January 11th 2012 contains 432,988 documents and is 3.6 Gigabytes in size.

**Table 1. Characteristics of the CJK Wikipedia collections used at NTCIR-10 Crosslink-2 task and for translation.**

| Language | Documents | Size | Dump Date |
|---|---|---|---|
| Chinese | 404,620 | 3.6GB | 11/01/2012 |
| Japanese | 858,610 | 9.8GB | 04/01/2012 |
| Korean | 297,913 | 2.2GB | 22/01/2012 |

### 2.2 Document Structure

Tags already present in the original Wikipedia XML dump files were maintained, but YAWN added new tags for article categories, sections, paragraphs, and links (amongst others). These new tags were added in an effort to provide additional structural information to the corpus user. The process followed has previously been successful at INEX. Examples of new and original tags, along with a brief description, are given in Table 2. The first column lists the tag, the second lists the source, and the third gives a brief description. For example, the title tag is original to the Wikipedia and gives the title of the article.

**Table 2. Example tags from the YAWN version of the Wikipedia dumps, the second column lists the source of the tag: W for Wikipedia and Y for YAWN**

| Tag | Source | Description |
|---|---|---|
| Title | W | Document title |
| Id | W | The document identifier |
| Link | Y | Used for hypertext links. Cross-language links contain an attribute (e.g. "xlink:label="ko"") giving the target language (one of: zh, ja, ko, en |
| timestamp | W | Last update timestamp |
| categories | Y | A list of categories |
| category | Y | An individual category (seen within the categories tag) |
| P | Y | Paragraph |
| Sec | Y | Article section |

An excerpt of YAWN processed article taken from the Chinese Wikipedia is show in Table 6. The article, 裹蒸 (a special type of Zongzi)[5] is showed as XML in the second row and as it appears in the Wikipedia in the fourth row. The other two rows are titles.

From the XML, it can be seen that the article is extensively marked up in XML and includes tags for such elements as: title, categories, and paragraphs. It also contains links to other Wikipedia articles.

## 3. CONVERSION TO ENGLISH

### 3.1 Considerations

Statistical machine translation systems are more effective when provided with context. That is, if asked to translate the contents of a short phrase, the accuracy can reasonably be expected to be lower than if given the entire sentence. Additionally, if the article was broken into its individual XML elements and each was translated separately it can reasonably be expected to take longer (using the Google API) than it would take for a single translation of the entire article.

However, the structural information has proven useful for both the presentation of articles and for providing *hint*s for the document retrieval. Wholesale removal of tags would detract from the utility of the translated collection and so as many tags as possible should be preserved through the translation. The preservation of tags also aids in the ability to map between the original and translated articles.

For the purpose of translating the collections herein, text formatting tags (b, i, li, etc.) and link tags (link, etc.) were removed before translation.

**Table 3. Sections removed from the Chinese articles before translation. No sections were removed from the Japanese or Korean articles**

| Section | Chinese |
|---|---|
| Notes | 注释<br>註釋 |
| References | 參考資料<br>参考资料<br>參考文獻<br>参考文献 |
| External Links | 外部連結<br>外部链接 |

To further increase throughput some sections were removed from the Chinese articles before translations. These sections included: “External Links”, “Notes”, and “References”. In each case the translation was likely to be of low accuracy and of little utility to the end user of the translated corpus.

[3] http://warehouse.ntcir.nii.ac.jp/openaccess/crosslink/10crosslink_documents.html

[4] http://ntcir.nii.ac.jp/CrossLink-2/

[5] http://en.wikipedia.org/wiki/Zongzi

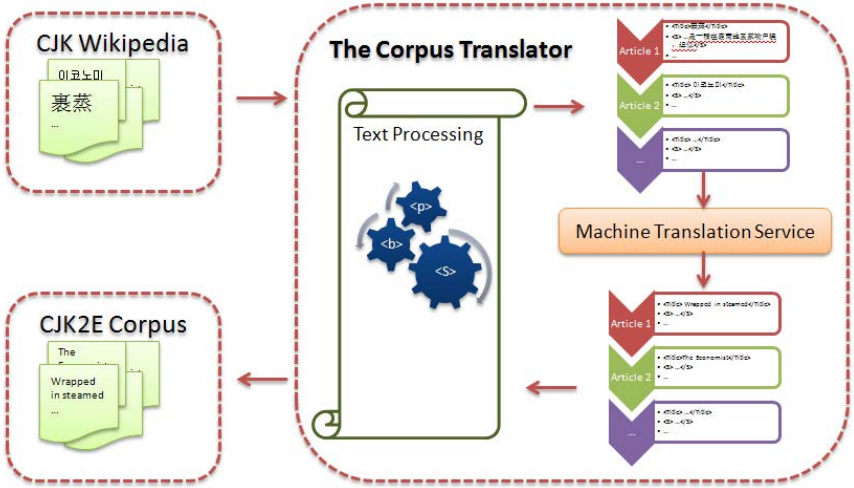



**Figure 1. System design of the corpus translator**

Table 3 presents a list of sections in column one and the Chinese equivalent in column two (including simplified, traditional, and other Chinese variants) removed from the documents before translation. For example, 注释 and 註釋 sections were removed as they were notes.

Such sections were only removed from the Chinese collections and not the Japanese or Korean collections because we did not have the necessary expertise to reliably identify such sections in those collections.

## 3.2 System Design

A design diagram of the corpus translator is given in Figure 1. The source article (in Chinese, Japanese, or Korean) is stripped of the formatting tags, and then decomposed into the remaining tags. They are further broken into sentences, which are segmented at the CJK punctuations where the segment size is not longer than 400 bytes (due to constraints imposed the translation service). Each chunk is translated and the translated chunks are re-composed into English articles and mark-up reintroduced.

## 3.3 The Statistics of the Translation

The translated corpus[6] contains 12,726,520 translated (to English) articles. Table 4 presents some statistics of the translated corpus. The first column lists the source language, the second gives the number of documents, and the third lists the size of the collection. For example, there are 397,571 articles translated from Chinese which take 2.6GB to store.

Not all articles successfully translated – but this is not unexpected. The translated corpus contains over twelve million articles that have been through YAWN conversion from Wikipedia formats into XML, processed by the corpus translator, translated by a third party, and pieced back together. Any one step in the process could reject a badly formed article or text segment. Version 1.0 of the collection consequently contains 98% of the original Chinese articles, less than 81% of the original Korean articles and only about 76% of the original Japanese articles. We are investigating the causes of the failures and will release updated collections in the future.

**Table 4. Characteristics of the translated collections**

| Language | Documents | Size |
|---|---|---|
| Chinese | 397,571 | 2.6GB |
| Japanese | 652,902 | 5.0GB |
| Korean | 239,285 | 1.3GB |

## 3.4 Translation Results

Table 7 shows a side by side comparison of the Chinese text and the translation results for the 裏蒸 article. The first column gives the name of the element, the second gives the original Chinese text and the third gives the English translation. For example, the title, 裏蒸, is translated as “Wrapped in steamed”.

It can be seen from the table (and from visual inspection of other translations) that short entity-like elements (title, category, etc.) have a high translation quality. For longer running text (paragraphs, etc.) the sentences contains many grammatical errors and sometimes make little sense. However, after reading the translation it is usually possible to understand the article (albeit with effort).

[6] http://www.clld.sef.qut.edu.au/corpus (to appear)

## 4. THE UESFULNESS OF THE CORPUS

The readability of the translated articles may be relatively poor because they were machine-translated and the machine translated article quality is not comparable to that of professionally translated article. Although the corpus is not created for human consumption, it may be still be useful despite the quality of machine translation. To demonstrate the usefulness of the CJK2E corpus, we designed a small experiment, which made use of this corpus for a cross-lingual link discovery (CLLD) task. The experiment compared two CLLD systems: the baseline system discovered links by searching the translated anchors in the target document collection; the other system found links through searching the anchors directly in the machine translated CJK2E corpus.

### 4.1 The Experiment

#### *4.1.1 Experiment Overview*

In NTCIR-9 CrossLink Task, there were 25 orphaned English Wikipedia articles used as test topics, and the metrics LMAP, R-Prec, and P@N were utilised for the CLLD system evaluation [1, 2]. A CLLD System is required to recommend prospective meaningful anchors for the test topics and for each anchor identify up to 5 relevant links in a different language. For each topic, up to 250 anchors are allowed.

Our experiment focused on the English-to-Chinese subtask. The document retrieval system employed for link retrieval was the ATIRE[7] open source search engine with its modified BM25 ranking function.

#### *4.1.2 Baseline Run*

Team QUT in the NTCIR-9 CrossLink task (English-to-Chinese) submitted a run (LinkProbIR_ZH) which first used a link mining method to recommend anchors in source documents, then translated those anchors into Chinese, and finally, searched the Chinese Wikipedia collection using the translated anchors as query terms for relevant links [5].

Their link discovery process is similar to a common approach to achieving cross-lingual information retrieval where queries are translated into the target language and then a monolingual IR system is used to locate the relevant documents in the target collection. This run will be used as the baseline run for system performance comparison.

#### *4.1.3 The Run with the CJK2E Corpus*

To compare the performance of the above system (that relies on translated anchors), an alternative run, LinkProbIR_CJK2E, was created by directly searching the anchor candidates in the CJK2E corpus. The number of identified links for each identified anchor was limited to 5, in accordance with the NTCIR-9 CrossLink task specification.

Trying to ensure that the two systems were comparable, the anchor candidates used to find links for the test topics in this run was the same as that used by the system to create run LinkProbIR_ZH. An anchor candidate (any given phrase), $\alpha$, is selected by measuring its anchor weight, $\gamma$ (>0) , [8] the probability of being an anchor. It is defined as:

$$\gamma = \frac{number\ of\ articles\ having\ (\alpha)\ being\ used\ as\ an\ anchor}{number\ of\ articles\ that\ have\ text\ of\ anchor(\alpha)} \quad (1)$$

[7] http://www.atire.org/

The anchor weight of anchor candidates was calculated using the data mined from the English Wikipedia corpus used in the Link-the-Wiki track of INEX [9, 10]. For each orphaned topic article, all possible n-gram substrings from the document were first computed. For each of these the $\gamma$ score was looked-up, and the list of anchor candidates was then sorted by their anchor weight values.

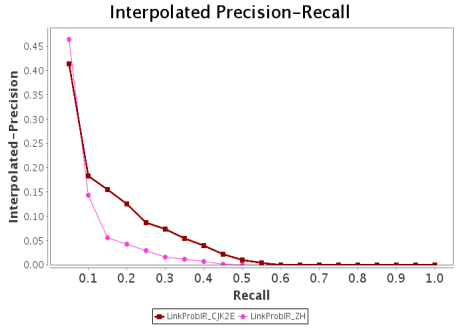



**Figure 2. The interpolated P-R curves of two systems**

### 4.2 Results and Discussions

The scores of the two experimental runs computed using the evaluation tool with the official *qrel* (Wikipedia ground-truth in file-to-file level) are given in Table 5. The runs are sorted on LMAP. For easy performance comparison, the interpolated precision and recall curves of two runs are also given in Figure 2.

From Figure 2, it can be seen that run LinkProbIR_CJK2E performs much better than run LinkProbIR_ZH. The scores computed in different metrics showed in Table 5 also suggest better links were discovered by run LinkProbIR_CJK2E than run LinkProbIR_ZH. Although, run LinkProbIR_ZH out-performs run LinkProbIR_CJK2E if measured with metrics P@5 and P@20, run LinkProbIR_CJK2E picked up many more good links if measured against a larger set of recommend anchors (P@30 or P@50, for example). A statistical analysis (two-tail paired *t*-test) on the LMAP scores of two runs over the 25 topics indicates that run LinkProbIR_CJK2E found significantly (p = 0.01) more links than LinkProbIR_ZH.

The experiment results indicate that the CJK2E corpus is useful in helping improve cross-lingual link discovery performance when cross-lingual information retrieval methods are involved. The performance difference of two systems could be attributed to the difference in translation quality. An article can provide more information to the translation engine than a single query with a few terms. The translation is expected to be superior as more contextual information is given. Translating a passage (as Google Translate does) is more likely to be correct than translating a word or a phrase at a time without the context of the embedding passage.

For CLLD approaches that first machine-translate anchor candidates into a target language and then searches them in the target document collection may discover fewer relevant links than other methods due to the possible inaccurate translation of *anchors* (caused by, for example, out of vocabulary (OOV)

terms). Although there are translation errors in the CJK2E corpus, overall most terms and phrases appear to have been correctly translated, resulting in an increased chance of hitting a relevant document if the anchors remain untranslated.

## 5. CONCLUSIONS

This article presents a machine-translated parallel English corpus which can be used by various cross-lingual link discovery, cross-lingual information retrieval, and machine learning systems to further improve their performance to satisfy users' information needs. An experiment was designed to justify the usefulness of the corpus, and the experimental results proved the claim. Wikipedia users may also find it useful because lots of articles can be adopted into the existing English Wikipedia with further proper editing and quality improvement.

## 6. ACKNOWLEDGMENTS

The Translate Research API is provided by Google.

## 7. REFERENCES

1. Tang, L.-X., Geva, S., Trotman, A., Xu, Y., Itakura, K.Y.: Overview of the NTCIR-9 Crosslink Task: Cross-lingual Link Discovery. In: Proceedings of NTCIR-9, pp. 437-463. (2011)

2. Tang, L.-X., Itakura, K.Y., Geva, S., Trotman, A., Xu, Y.: The Effectiveness of Cross-lingual Link Discovery. In: Proceedings of The Fourth International Workshop on Evaluating Information Access (EVIA), pp. 1-8. (2011)

3. Knoth, P., Zilka, L., Zdrahal, Z.: KMI, The Open University at NTCIR-9 CrossLink. In: Proceedings of NTCIR-9, pp. 495-502. (2011)

4. Kim, J., Gurevych, I.: UKP at CrossLink: Anchor Text Translation for Cross-lingual Link Discovery. In: Proceedings of NTCIR-9, pp. 487-494. (2011)

5. Tang, L.-X., Cavanagh, D., Trotman, A., Geva, S., Xu, Y., Sitbon, L.: Automated Cross-lingual Link Discovery in Wikipedia. In: Proceedings of NTCIR-9, pp. 512-519. (2011)

6. Fahrni, A., Nastase, V., Strube, M.: HITS' Graph-based System at the NTCIR-9 Cross-lingual Link Discovery Task. In: Proceedings of NTCIR-9, pp. 473-480. (2011)

7. Schenkel, R., Suchanek, F., Kasneci, G.: YAWN: A Semantically Annotated Wikipedia XML Corpus. In: 12. GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW 2007). (2007)

8. Itakura, K., Clarke, C.: University of Waterloo at INEX2007: Adhoc and Link-the-Wiki Tracks. In: Proceedings of INEX 2007, pp. 417-425. (2008)

9. Huang, W., Geva, S., Trotman, A.: Overview of the INEX 2009 Link the Wiki Track. In: Proceedings of INEX 2009, pp. 312-323. Springer Berlin / Heidelberg, (2010)

10. Trotman, A., Alexander, D., Geva, S.: Overview of the INEX 2010 Link the Wiki Track In: Proceedings of INEX 2010, pp. 241-249. Springer Berlin / Heidelberg, (2011)

**Table 5. The F2F automatic evaluation scores of two experimental runs with metrics (LMAP, R-Prec, and P@N)**

| Run ID | LMAP | R-Prec | P5 | P10 | P20 | P30 | P50 | P250 |
|---|---|---|---|---|---|---|---|---|
| LinkProbIR_CJK2E | 0.044171 | 0.119954 | 0.128 | 0.152 | 0.138 | 0.141333 | 0.136 | 0.07536 |
| LinkProbIR_ZH | 0.02338 | 0.067135 | 0.184 | 0.16 | 0.118 | 0.109333 | 0.084 | 0.04352 |

**Table 6. Excerpt from the Chinese Wikipedia article 裹蒸 about a special kind of Zongzi. The top shows the YAWN XML output and the bottom shows the original text as it appears in the Wikipedia**

**The article XML file**

```
...
<title>裹蒸</title>
<categories>
<category>:點心</category>
<category>:糯米食品</category>
</categories>
<bdy>
<b>裹蒸</b>，俗作<b>裹蒸粽</b>，為<link xlink:type="simple" xlink:href="../../pages/263/263.xml">
中國</link><link xlink:type="simple" xlink:href="../../pages/412/412.xml">
廣東省</link><link xlink:type="simple" xlink:href="../../pages/813/15813.xml">
肇慶市</link>的特產，是一種在嶺南地區家喻戶曉，近似<link xlink:type="simple" xlink:href="../../pages/429/40429.xml">
粽</link>的煮食方式。傳統的裹蒸粽用料是<link xlink:type="simple" xlink:href="../../pages/266/39266.xml">
糯米</link>、<link xlink:type="simple" xlink:href="../../pages/880/372880.xml">
綠豆</link>、半肥瘦<link xlink:type="simple" xlink:href="../../pages/024/111024.xml">
豬肉</link>作餡，必须用西江两岸特有的<link>
冬葉</link>来包裹蒸制，具有独特的清香与优良的防腐作用。現在的裹蒸粽餡料包括五香肥肉、<link xlink:type="simple" xlink:href="../../pages/196/300196.xml">
鹹蛋</link>黃、燒<link xlink:type="simple" xlink:href="../../pages/554/15554.xml">
雞</link>、<link xlink:type="simple" xlink:href="../../pages/589/277589.xml">
燒鴨</link>、<link xlink:type="simple" xlink:href="../../pages/744/100744.xml">
叉燒</link>等。<p>
...
```

**The real page on Chinese Wikipedia site**

裹蒸 [编辑]

维基百科，自由的百科全书

（重定向自裹蒸粽）

**裹蒸**，俗作**裹蒸粽**，為中國廣東省肇慶市的特產，是一種在嶺南地區家喻戶曉，近似粽的煮食方式。傳統的裹蒸粽用料是糯米、綠豆、半肥瘦豬肉作餡，必須用西江兩岸特有的冬葉来包裹蒸制，具有独特的清香与优良的防腐作用。現在的裹蒸粽餡料包括五香肥肉、鹹蛋黃、燒雞、燒鴨、叉燒等。

南齊書有記載：在皇帝御食中，有"裹蒸"這一道美食。皇帝稱：「我食此不盡，可留片破之，剩餘供當晚食。」裹蒸

**Table 7. A side by side comparison of the original Chinese text and the translation for the article 裹蒸（"wrapped in steamed")**

| Element | Chinese Source Text | English Translated Text |
|---|---|---|
| title | 裹蒸 | Wrapped in steamed |
| category | 點心 | Dim Sum |
| category | 糯米食品 | Glutinous rice food |
| abstract | 裹蒸，俗作裹蒸粽，為中國廣東省肇慶市的特產，是一種在嶺南地區家喻戶曉，近似粽的煮食方式。傳統的裹蒸粽用料是糯米、綠豆、半肥瘦豬肉作餡，必須用西江两岸特有的冬葉来包裹蒸制，具有独特的清香与优良的防腐作用。現在的裹蒸粽餡料包括五香肥肉、鹹蛋黃、燒雞、燒鴨、叉燒等。 | Wrapped in steamed, popular for Guo Zhengzong, Guangdong, China Zhaoqing City's specialty, is a Lingnan region well-known approximate dumplings of cooking way. Traditional Guo Zhengzong materials is glutinous rice, mung bean, half fat and lean pork for filling, must use West River cross-strait unique winter leaves to parcel steaming, with a unique fragrance and excellent anti-corrosion role now Guo Zhengzong filling spiced fat salted egg yolk, burning chicken.Roast duck, barbecued pork. |
| p | 南齊書有記載：在皇帝御食中，有“裹蒸”這一道美食。皇帝稱：「我食此不盡，可四片破之，剩餘權當晚食。」裹蒸是當地人作為春節、端午節探親拜年，必備的禮品，寓意豐衣足食和來年好運。清代詩人王仕禎有詩贊肇慶城鄉除夕熬煮裹蒸的盛況：「除夕濃煙籠紫陌，家家廛甑裹蒸香。」如今在肇慶的大街小巷中，觸目皆為肇慶裹蒸，與端硯齊名。不過，到現在的裹蒸依然採用人手製作。 | Nan Qi Shu have recorded: the emperor imperial food, there are "wrapped in steamed" This Road food. Emperor said: "I eat this endless, four destructible, the remaining right to the night food. Wrapped in steamed locals as the Spring Festival, Dragon Boat Festival to visit relatives New Year, essential gift, symbolize affluence and the coming year, good luck. Qing Dynasty poet Raymond Wong Zhen poetry praise Zhaoqing urban and rural New Year's Eve boiled wrapped in steamed the grand occasion:" New Year's Eve smoke cage purple street, every household dust steamer wrapped steamed fragrant. "Now in theThe streets in Zhaoqing, Zhaoqing wrapped in steamed caught the attention of all and Duan par. However, now wrapped in steamed still handmade. |
| p | 肇庆人制作裹蒸粽的传统从秦代就有了，其起源流传着两种说法，一种是秦军征统岭南时期，为驻军而设的可随身携带食用的煮熟的叶裹米团，一种是当时的农民在田间做活时，随身携带煮熟了的叶裹米团以便随时食用，但无论如何，裹蒸的来源并不是一般认为纪念屈原的端午粽，在肇庆民间一直是春节过年的食品，这就是裹蒸和粽子的区别。 | The Zhaoqing person production Guo Zhengzong the tradition From the Qin Dynasty, there had its origin spread with two arguments, one kind is the period of Qin Zheng unified Lingnan for the garrison and set the can carry carry consumption of cooked leaves wrapped m group, one kind is time farmers in the field QUICKER player to carry cooked the leaves wrapped m group so that at any time eating, and in any case, but, wrapped in steamed the source is not generally think to commemorate Qu Yuan Dragon Boat Rice Dumplings Zhaoqing Folk Spring Festival, Chinese New Year's food, wrapped in steamed and dumplings the difference. |
| section title | 著名品牌 | Well-known brand |
| p | 肥仔偉：一個旅行團到訪當地經常都會拜訪的裹蒸生產商。 | Fatty Wei: a tour group visited the local often visit wrapped in steamed manufacturer. |

# Exploiting Medical Hierarchies for Concept-based Information Retrieval

Guido Zuccon[1], Bevan Koopman[1,2], Anthony Nguyen[1], Deanne Vickers[1], Luke Butt[1]

[1]Australian e-Health Research Centre, CSIRO, Brisbane, Australia

[2]Faculty of Science & Technology, Queensland University of Technology, Brisbane, Australia

{guido.zuccon, bevan.koopman, anthony.nguyen, deanne.vickers, luke.butt}@csiro.au

## ABSTRACT

Search technologies are critical to enable clinical staff to rapidly and effectively access patient information contained in free-text medical records. Medical search is challenging as terms in the query are often general but those in relevant documents are very specific, leading to granularity mismatch.

In this paper we propose to tackle granularity mismatch by exploiting subsumption relationships defined in formal medical domain knowledge resources. In symbolic reasoning, a subsumption (or 'is-a') relationship is a parent-child relationship where one concept is a subset of another concept. Subsumed concepts are included in the retrieval function. In addition, we investigate a number of initial methods for combining weights of query concepts and those of subsumed concepts. Subsumption relationships were found to provide strong indication of relevant information; their inclusion in retrieval functions yields performance improvements. This result motivates the development of formal models of relationships between medical concepts for retrieval purposes.

## Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.3 Information Search and Retrieval

## General Terms

Theory, Experimentation

## Keywords

Medical Information Retrieval, Subsumption, SNOMED CT

## 1. INTRODUCTION

Search technologies that enable clinical staff to rapidly and effectively search patients health records may improve health outcomes as well as produce time and costs savings [3]. However, searching medical records can be challenging: keyword based approaches often fail to identify medical entities that are referred to with different terms, such as the synonymous terms 'heart attack' and 'myocardial disorder'. Concept-based retrieval approaches have been proposed to overcome keyword search problems [5]. In these approaches, the original free-text documents are converted to concepts defined in medical ontologies, such as the SNOMED CT ontology.

Mismatch in granularity between concepts in a query and those found in relevant documents may however hinder retrieval effectiveness. For example, a medical record document may contain detailed notes about the brand and dosage of drugs prescribed to a patient, whereas a query would contain only the general class of drugs or its active ingredient. Previous concept-based approaches are susceptible to granularity mismatch.

Within ontologies, concepts are organised in inheritance hierarchies, with parent-child, or *subsumption*, relationships. For example, the hierarchy for *Opiate* in the SNOMED CT ontology is shown in Figure 1. The figure shows that the different types of Opiate are subsumed by the parent Opiate. In a retrieval scenario, documents that contained these subsumed concepts would likely be relevant to a query that contains Opiate. Subsumption relationships are not accounted for in most current concept-based approaches for medical records information retrieval; successful use of subsumption has been shown in related domains, e.g. [2].

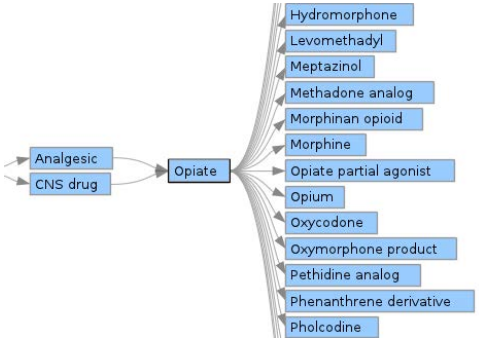



**Figure 1: SNOMED CT hierarchy for the class of drug *Opiate*.**


*ADCS'12*, December 05 – 06 2012, Dunedin, New Zealand.

We hypothesise that accounting for subsumption between concepts in retrieval methods may allow for higher effectiveness in medical search. To this end, we provide an initial empirical investigation of the use of subsumption to enhance medical information retrieval. In the experiments, we consider the scenario modelled by the TREC Medical Records Track, where a health practitioner consults a collection of electronic health records to individuate cohorts suitable for participating to a clinical study.

Subsumption information is taken from the SNOMED CT hierarchy and included in the retrieval function. In addition, a number of initial methods for combining weights of query concepts and those of subsumed concepts are evaluated. Empirical results demonstrate that subsumed concepts provide useful information that can be used to improve retrieval effectiveness.

## 2. METHODS

Following the work of Koopman et al. [4, 5], we implement a 'bag-of-concepts' representation of documents rather than the traditional bag-of-words. Terms are transformed to concepts using the natural language processing tool Metamap; concepts are derived from the SNOMED CT ontology.

Documents are scored according to (1) the weight of query concepts in a document, and (2) the weight of concepts in a document that have been subsumed by a query concept. For each query concept $c_i$ we obtain the list of subsumed concepts $c_j \prec c_i$ from the SNOMED CT ontology. These subsumed concepts are included in the retrieval function, leading to the following retrieval status value (RSV):

$$RSV(d|q) = \sum_{c_i \in q} w(c_i, d) + \sum_{c_j \prec c_i; c_i \in q} \delta(w(c_j, d)) \quad (1)$$

where $w(c_i, d)$ is the weight of concept $c_i$ in document $d$, and $\delta(w(c_j, d))$ adjusts the weight of a subsumed concept $c_j$. That is, the score of a document for a query $q$ is the sum of the weights associated with the query concepts and the adjusted weights of the concepts that are subsumed by the query concepts.

Equation 1 is a general method to integrate subsumed concepts into the retrieval function. A number of instantiations of both $w(c_i, d)$ and $\delta(w(c_j, d))$ are possible.

In the following we outline some possible variations of both type of functions; these are then empirically evaluated in Section 3.

### 2.1 $\mathbf{w(c_i, d)}$ : Weighting Concepts

Next, we consider a number of possible instantiations of the weighting function $w(c_i, d)$. As overarching weighting schema, we used variations of tf-idf as Koopman et al. found that in the medical domain this often yields higher retrieval performance than alternative approaches, such as BM25 and language models [5].

**cfidf:** this corresponds to the normalised tf-idf weighting schema[1], where concepts are used instead of terms, i.e.:

$$w(c_i, d)_{\text{cfidf}} = \frac{count(c_i, d)}{l_d} \cdot \log \frac{|D|}{|d(c_i)|} \quad (2)$$

and $count(c_i, d)$ is the frequency of concept $c_i$ in document $d$, $l_d$ is the length of document $d$, $|D|$ is the total number of documents in the collection and $|d(c_i)|$ is the number of documents that contain concept $c_i$.

**ncfidf:** in this instantiation a concept frequency is normalised by its frequency in the collection (i.e. the maximum likelihood estimation is used for the concept frequency component), i.e.:

$$w(c_i, d)_{\text{ncfidf}} = \frac{count(c_i, d)}{count(c_i)} \cdot \log \frac{|D|}{|d(c_i)|} \quad (3)$$

and $count(c_i)$ is the frequency of concept $c_i$ in the collection.

**ecfidf:** this corresponds to the enhanced tf-idf described by Zhai [8] in which the Okapi formula is used for weighting term frequencies and where concepts are used instead of terms, i.e.:

$$w(c_i, d)_{\text{ecfidf}} = \frac{k_1 count(c_i, d)}{count(c_i, d) + k_1(1 - b + b\frac{l_d}{l_{avg}})} \cdot \log \frac{|D|}{|d(c_i)|} \quad (4)$$

and $l_{avg}$ is the average document length, and $k_1$, $b$ are the Okapi parameters.

### 2.2 $\delta(\mathbf{w(c_j, d)})$: Integrating Subsumption

Next, we consider how the weight of a concept should be adjusted if it was subsumed by the query.

A straightforward approach would to treat subsumed concepts in the same way as query concepts, i.e. $\delta(w(c_j, d)) = w(c_j, d)$. We call this approach linear.

However, the presence of a subsumed concept in a document may offer a different indication of relevance than an actual query concept. A subsumed concept indicates a specialisation of the parent concept, and thus treated differently to an actual query concept. Intuitively a subsumed concept would be a weaker indication of relevance than a query concept. Alternatively, a subsumed concept may be a stronger indication of relevance because it is an actual specialisation of the more general concept used in the query as it is more focused and less ambiguous. To this end, we consider a number of instantiations of $\delta(w(c_j, d))$ that encompass the two alternative rationales.

**sqrt($\mathbf{w(c_j, d)}$):** the weight for the subsumed concept $c_j$ in the document is adjusted according to the square root of the weight $w(c_j, d)$, i.e.:

$$\delta(w(c_j, d)) = \sqrt{w(c_j, d)} \quad (5)$$

In this case a subsumed concept contributes less evidence towards the score of a document than a query concept.

**log($\mathbf{w(c_j, d)}$):** the weight for the subsumed concept $c_j$ in the document is the logarithm of the weight $w(c_j, d)$, i.e.:

$$\delta(w(c_j, d)) = \log[w(c_j, d)] \quad (6)$$

If $w(c_j, d)$ is less than one, then the subsumed concept receives a negative weight[2]. In this case, the weight of a subsumed concept but be considerably higher than that of a query concept to influence the score.

[1]Note, the standard tf-idf weighting (no document length normalisation) performed significantly worse.

[2]We excluded the case $w(c_j, d) = 0$ to avoid $\log[w(c_j, d)] = -\infty$; in this case a zero weight is assigned to log($\mathbf{w(c_j, d)}$).

**pow($\mathbf{w(c_j, d)}$):** the weight for the subsumed concept $c_j$ in the document is the square of the weight $w(c_j, d)$, i.e.:

$$\delta(w(c_j, d)) = [w(c_j, d)]^2 \quad (7)$$

In this instantiation, more weight (and thus importance) is given to subsumed concepts rather than query concepts.

**exp($\mathbf{w(c_j, d)}$):** the weight for the subsumed concept $c_j$ in the document is the (natural) exponential function of the weight $w(c_j, d)$, i.e.:

$$\delta(w(c_j, d)) = e^{w(c_j, d)} \quad (8)$$

Here, subsumed concepts become the main influence of the document's score.

# 3. EMPIRICAL EVALUATION

## 3.1 Baselines

To understand the empirical merits of using subsumption information to retrieve medical documents, we compare approaches that score query concepts and their subsumed concepts against approaches that do not consider subsumed concepts. The baseline using no subsumption is indicated no sub., i.e. $\delta(w(c_j, d)) = 0$. Where applicable, parameters were set to the common Okapi values[3].

## 3.2 Test Collection

We use the TREC 2011 Medical Records Track, a collection of 100,866 clinical record documents taken from U.S. hospitals. Documents belonging to a single patient's admission were concatenated together into a single document called a patient visit document; this is consistent with the unit of retrieval used TREC 2011 MedTrack and collapsing reports to patient visits was a common practise among many TREC MedTrack systems[4].When documents are grouped into visits, the corpus then contains 17,198 patient visit documents.

| Corpus | #Docs | Avg. doc. len. | #Vocab. |
|---|---|---|---|
| MedTrack: | | | |
| Terms | 17,198* | 2338 terms/doc | 218,574 |
| Concepts | 17,198* | 6066 concepts/doc | 54,143 |

*100,866 original reports collapsed to 17,198 patient *visit* documents.

**Table 1: Collection statistics for the TREC MedTrack'11 corpus of clinical records. Statistics are provided for the original term corpus and subsequent corpus after conversion to SNOMED CT concepts.**

The original free-text documents were translated into concept identifiers from the SNOMED CT medical terminology using the information extraction system MetaMap, as suggested by Koopman et al. [5]. Statistics for both the term and concept corpora are provided in Table 1.

The 34 topics from the TREC MedTrack'11 collection were used in the experiments. Retrieval results were evaluated using Bpref and Precision @ 10 in accordance with TREC MedTrack'11. Because the absolute number of judged documents per topic is small, the computation of metrics such as MAP, nDCG, etc. is not meaningful.

## 3.3 Results

Table 2 outlines the results of the investigated approaches. Results show the effect of different combinations of methods for weighting concepts and adjust the weights of subsumed concepts. For each concept weighting method, the best performances are highlighted in bold. No statistical significant differences are measured between variations of $\delta(w(c_j, d))$ and the corresponding baselines (i.e. no sub.)

Further discussion of the results obtained when considering the concept-based representation and subsumption follows in the next section.

# 4. CONTRIBUTION OF SUBSUMPTION

The empirical results demonstrate that subsumption relationships supply strong relevant information that can lead to effective retrieval performance.

The use of only subsumed concepts to score documents (sub. only), thereby ignoring matching the query concepts, obtains mixed results based on the employed weighting schemas. However, none of these sensibly improve the corresponding concept baseline.

It is instead when the contribution of subsumed concepts is combined with that of matching query concepts that promising improvements of retrieval performance are witnessed. Specifically, ecfidf used in combination with sqrt($\mathbf{w(c_j, d)}$) yields the best Bpref values in our experiments. Whereas, ecfidf used in combination with the linear approach yields the highest P@10.

However, no one approach is found that performs the best across the different weighting method $w(c_i, d)$. For example, while using exp($w(c_j, d)$) to weight subsumed concepts obtained the best retrieval performance with cfidf, results obtained with other instantiations of $w(c_i, d)$ do not follow this trend. In particular, when ecfidf is considered, increasing the subsumed concepts' weights using the exponential function actually yields lower B-pref and P@10 than all the other subsumed concept weighting methods.

When ncfidf and ecfidf are considered, both the linear and sqrt($w(c_j, d)$) approaches for adjusting the weights of subsumed concepts yield improvements over the respective concept baselines[5]. But the best function to apply to adjust the weights of subsumed concepts is unclear.

Other approaches for $\delta((w(c_j, d))$ perform lower than the concept baseline (no sub.), with the exception of log($\mathbf{w(c_j, d)}$) when ecfidf is used: performance increments here are however minimal.

Parallels can be drawn between our approaches, that combine query concepts and subsumed concepts, and the query expansion process [1]. These are similar because they both

[3] $b = 0.75$, $k_1 = 1.2$

[4] http://trec.nist.gov/pubs/trec20/t20.proceedings.html

[5] Except the combination of ecfidf and simple, which yields a Bpref lower than that of the concept baseline.

| | | | $\delta(\mathbf{w}(\mathbf{c_j}, \mathbf{d}))$ | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | no sub. | sub. only | linear | sqrt($\mathbf{w}(\mathbf{c_j}, \mathbf{d})$) | log($\mathbf{w}(\mathbf{c_j}, \mathbf{d})$) | pow($\mathbf{w}(\mathbf{c_j}, \mathbf{d})$) | exp($\mathbf{w}(\mathbf{c_j}, \mathbf{d})$) |
| $\mathbf{w}(\mathbf{c_i}, \mathbf{d})$ | cfidf | .3943 | .2002 | .4080 | .4216 | .3805 | .3791 | **.4330** |
| | | .2500 | .3147 | .3088 | .3647 | .3500 | .2324 | **.4206** |
| | ncfidf | .4430 | .4440 | **.4544** | .4447 | .3831 | .4440 | .4296 |
| | | .3765 | .3765 | .4265 | **.4353** | .3441 | .3765 | .4176 |
| | ecfidf | .4799 | .4691 | .4789 | **.4814** | .4800 | .4691 | .4469 |
| | | .4941 | .4265 | **.5147** | .5029 | .5000 | .4265 | .3118 |

**Table 2: Results obtained by the weighing approaches on TREC MedTrack'11, where $\mathbf{w}(\mathbf{c_i}, \mathbf{d})$ refers to instantiation of the weighting function for query concepts, and $\delta(\mathbf{w}(\mathbf{c_j}, \mathbf{d}))$ refers to instantiations of the weighing function for concepts subsumed by query concepts. For each weighting schema, the first row of results reports the measured Bpref values; the second row reports the corresponding P@10 values. The column labelled no sub. reports the performance of the approaches that do not consider subsumed concepts. The column labelled sub. only refers to results obtained when weighting subsumed concepts only, thus ignoring query concepts.**

score documents against the original query and an additional set of terms (concepts) derived from the initial request. However, most query expansion approaches do not weight the expanded terms; weighted query expansion is less common than its unweighted version. In addition, most query expansion techniques rely on corpus statistics to select candidate terms for expansions, and a threshold or limit on number of candidate terms is usually employed. In the approaches proposed in this paper, instead, concepts other than those in the query are selected because their relationship with a query concept present in a document is formally encoded in a domain knowledge source. Corpus statistics are thus used for the weighting process, not for the selection process. In addition, no limit is imposed on the number of additional concepts that are considered when scoring documents, the number of additional concepts is taken from the number of subsumed concepts for a query concept.

## 5. CONCLUSIONS

This work is an initial investigation on the use of subsumption for concept-based medical information retrieval. Empirical results have shown potential increase in retrieval performance when considering the matching between documents and subsumed concepts alongside with query concepts. The approaches investigated in this paper were based on functions that combine weights of query concepts with those of subsumed concepts; functions that adjust the latter weights were also explored. The best performance was highly dependent either on the specific tf-idf variation considered, or on the specific function used to distinguish the contribution of subsumed concepts, or both. No single approach has provided strong, consistent gains over the concept baselines. How to best combine weights for query concepts and subsumed concepts is an open line of research, but this paper demonstrated promising initial results.

Future work will be directed towards the creation of formal models able to capture the two different matching mechanisms. Specifically, these models may take advantage of additional information regarding the subsumption concepts, for example the distance between the child subsumed concept and the parent concept or the extent the concepts are semantically related [7, 6].

## 6. REFERENCES


[1] C. Carpineto and G. Romano. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)*, 44(1):1, 2012.

[2] M. Douyère, L. Soualmia, A. Névéol, A. Rogozan, B. Dahamna, J. Leroy, B. Thirion, and S. Darmoni. Enhancing the MeSH thesaurus to retrieve French online health resources in a quality-controlled gateway. *Health Information & Libraries Journal*, 21(4):253–261, 2004.

[3] W. Hersh. *Information retrieval: a health and biomedical perspective*. Springer Verlag, New York, 3rd edition, 2009.

[4] B. Koopman, P. Bruza, L. Sitbon, and M. Lawley. AEHRC & QUT at TREC 2011 Medical Track : a concept-based information retrieval approach. In *20th Text REtrieval Conference (TREC 2011)*, pages 1–7, Gaithersburg, MD, USA, Nov. 2011. NIST.

[5] B. Koopman, P. Bruza, L. Sitbon, and M. Lawley. Towards Semantic Search and Inference in Electronic Medical Records: an approach using Concept-based Information Retrieval. *Australasian Medical Journal: Special Issue on Artificial Intelligence in Health*, 5(9):482–488, 2012.

[6] B. Koopman, G. Zuccon, P. Bruza, L. Sitbon, and M. Lawley. An Evaluation of Corpus-driven Measures of Medical Concept Similarity for Information Retrieval. In *21st ACM International Conference on Information and Knowledge Management (CIKM)*, Maui, USA, 2012.

[7] T. Pedersen, S. V. S. Pakhomov, S. Patwardhan, and C. G. Chute. Measures of semantic similarity and relatedness in the biomedical domain. *Journal of biomedical informatics*, 40(3):288–99, June 2007.

[8] C. Zhai. Notes on the Lemur TFIDF model. Technical report, School of Comp. Sci., CMU, 2001.

# Finding Additional Semantic Entity information for Search Engines

Jun Hou
Queensland University of Technology
Brisbane
jun.hou@student.qut.edu.au

Richi Nayak
Queensland University of Technology
Brisbane
r.nayak@qut.edu.au

Jinglan Zhang
Queensland University of Technology
Brisbane
jinglan.zhang@qut.edu.au

## ABSTRACT
Entity-oriented search has become an essential component of modern search engines. It focuses on retrieving a list of entities or information about the specific entities instead of documents. In this paper, we study the problem of finding entity related information, referred to as *attribute-value pairs*, that play a significant role in searching target entities. We propose a novel decomposition framework combining reduced relations and the discriminative model, Conditional Random Field (CRF), for automatically finding entity-related attribute-value pairs from free text documents. This decomposition framework allows us to locate potential text fragments and identify the hidden semantics, in the form of attribute-value pairs for user queries. Empirical analysis shows that the decomposition framework outperforms pattern-based approaches due to its capability of effective integration of syntactic and semantic features.

## Categories and Subject Descriptors
I.2.7 [**Computing Methodologies**]: Natural Language Processing – *Language parsing and understanding; Text analysis*

## General Terms
Algorithms, Design, Experimentation.

## Keywords
Entity Retrieval, Decomposition Framework, Conditional Random Field (CRF)

## 1. INTRODUCTION
Due to the rapidly increasing size and wide spread of the Web, it has become an immense knowledge repository which contains rich information of entities and their relations. In parallel, as search engine technologies evolve, Entity Retrieval [1, 21] and Question Answering [26] have become crucial components of modern web information retrieval systems. Entity Retrieval and Question Answering seek to find information of individual entities that meet the expected constraints imposed by users in the form of queries, rather than the documents. The constraints help filter out irrelevant answer candidates and determine the answer selection criteria.


*ADCS '12*, December 5 – 6, 2012, Dunedin, New Zealand.


Understanding search constraints at entity level in Web queries has been a focused theme of many research methods from the area of Query Intent Classification [3, 14, 22], Query Modifier identification [18, 19] and Identifying Semantic Structure of Queries [15]. These methods are largely driven by the requirement of entity level searchable information. On the other hand, ontologies (with knowledge on entity level) have been used to boost the performance of entity-oriented search. Several lightweight ontologies that encode hierarchical entity related information have been proposed including Yago [24], Freebase[1], amd DBpeida[2]. Generally, these ontologies contain Class-Instance and Class-Attribute hierarchy at entity level and non-taxonomic entity relations such as "hasWonPrize". Wikipedia category is used in [25, 28], Yago has been used in [7] and Freebase is used in [6]. However, it is very difficult to find a comprehensive ontology that covers all entity related information for a general domain. For example, Yago and DBpedia target a limited number of non-taxonomic entity relations because of using handcrafted rules [4]. On the other hand, free text in documents holds rich context and linguistic information and contains entity level information implicitly.

In this paper, we focus on the problem of automatically finding semantic information for entities by integrating linguistic information and external domain knowledge. Our goal is to identify all potential entity level information, such as ontology-like Class-Attribute and any non-taxonomic relations for a target entity in free text. Web search engines can then provide more precise results based on the fine-grained semantic information about entities instead of just returning documents based on keywords matching.

Unlike structured data sources, the entity related information in free text is usually formulated as sequences of words without much explicit semantic information. In our work, entity related information is modeled as *attribute-value pairs*, (<*attribute*>, <*value*>). Traditionally, a set of entities consists of an entity class exhibiting a set of properties. These general entity properties inherited from the entity class can be referred to as "*noun phrase attributes*". However, unlike the pairs of *Class-Attribute* are explicitly represented in domain ontology, attribute-value pairs in free text often exist in an implicit form. For example, the fragment "Australian state of Victoria", does not contain any segment that corresponds to the attribute name "country" for its value "Australian". A significant sub-task presented in this paper is identifying the explicit and implicit attributes and their values. Moreover, the attribute-value pair can be described in the form of

1 http://www.freebase.com/

2 http://dbpedia.org/About

non-taxonomic relation, which we refer to as "*relation attribute*". For example, the relation "FoundDate" with its value "1785" can be interpreted from the text "Melbourne is founded in 1785". This type of relation plays a significant role for answering factual questions. For example, identification of a relation between recording company and Kingston Trio's songs, would be vital to answer the query such as "What recording companies now sell the Kingston Trio's songs?".

In this paper, we propose a decomposition framework by reducing the triple that encodes the relations between attribute-value pairs and entities. The triple $r = < entity,\ attribute,\ value >$ is reduced to $r' = < entity,\ class >$ and $r'' = < entity|class,\ value|attribute - value >$. (The "$entity|class$" in $r''$denotes that either $entity$ or $class$ can be the left argument. Similarly $value|attribute - value$ denotes that either value or attribute-value can be the right argument showing that it is possible to have implicit or explicit attribute for an entity's value.). The property-denoting attribute-value pairs can be inferred by finding the reduced relations $r'$ and $r''$, and then by identifying the semantic roles in $r'$ and $r''$, i.e., $entity, class, attribute\ and\ value$.

In the decomposition framework, the reduced relation $r'$ is first detected from the context text. Once the relation between entity and its class is identified, the task is extended to identify $r''$ for finding attribute-value pairs. We then propose to apply Conditional Random Fields (CRF) models to assign semantic role for elements in $r'$ and $r''$, i.e., $entity, class, attribute\ and\ value$. Both noun phrase attribute and relation attribute will be found in $r'$ and $r''$.

More specifically, contributions of our work are as follows: (1) Modeling entity level information as attribute-value pairs; (2) Proposing a novel decomposition framework for automatically finding attribute-value pairs; and (3) Presenting methods that identify semantics of attribute-value pairs with the related entity.

Section 2 discusses related work. Section 3 introduces the proposed decomposition framework. Evaluation is presented in Section 4. Section 5 concludes this paper and discusses future work.

## 2. RELATED WORK

General relations containing class-attribute and the associated entity properties are valuable for building concept representations. The framework proposed in this paper integrates such upper level knowledge as features for finding all possible attribute-value pairs of an entity. The majority of existing entity property studies uses a semi-supervised learning approach to extract class-attribute pairs for an entity [16, 17]. These methods aim to extract general class-attribute information, i.e. finding attributes like, "director" or "cast" for an entity class, "movie". Researchers have identified class instances (entities) from unstructured text with seed entities and use them to extract attributes from query logs using query templates. In order to provide high coverage and quality class-attribute, lightweight ontologies, such as YAGO [24] and DBpedia have been developed to integrate entity level information from different sources. These ontologies consist of entities grouped into different entity classes and each entity is attached to related attributes and relations. However, these works mainly focus on noun phrase attribute and target a limited number of pre-defined relations.

Another relevant research area to our work of studying semantics of attribute-value pairs in free text is the semantic studies of adjective-noun phrases, i.e., assigning attributes to property-

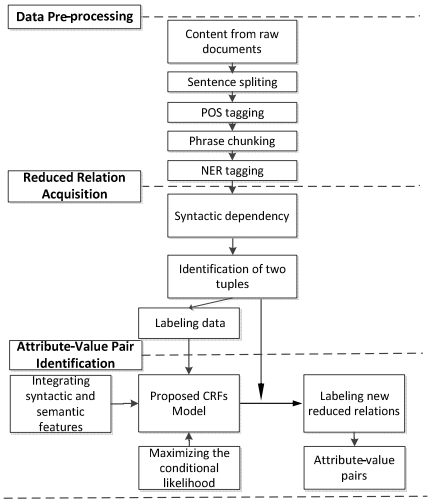



**Figure 1. Overview of our proposed decomposition framework.**

denoting adjectives. For example, in "a blue car", the hidden attribute "color" should be assigned to the value "blue". In particular, authors in [11] developed a representation composition framework and utilized structured vector space models (SVSM) to map adjective-noun phrases to attribute semantics. They [12] further proposed an approach using Topic Models of LDA to discover the inherent semantics between the attribute and the adjective. Authors in [10] leverage the Expectation-Maximization (EM) algorithm to learn an attribute-value classifier for a similar task. A key difference with our work from these works is that we extend the attribute-value pair identified from the noun phrase attribute to the relation attribute and focus on connecting attribute-value pairs with the corresponding entity. For instance, we identify the attribute "color" for the value "blue" as well as which entity is related to "a blue car".

Another research area that is related to this paper is Open Information Extraction (IE), as finding attribute-value pairs for an entity can be viewed as a specific task discovering reduced relations between attribute-value pairs and entity without pre-defined rules. The task of Open IE was introduced by [8] with a state-of-the-art Web IE system, TEXTRUNNER. The system learns unknown relations based on self-supervised framework using a small set of domain-independent features from the training set. This framework is further extended to utilize different types of CRF such as supervised, self-supervised and stacked for extracting relations [5]. [27] proposes a novel Open IE system based on syntactic dependency representation using the structured sources from Wikipedia Infobox. Second generation Open IE systems, such as Reverb [20] and R2A2 [9] are proposed to further improve extraction performance. The differences between these works and ours are that the open IE systems only deal with explicitly mentioned relations, whereas, we focus on finding all possible reduced relations as well as finding the attribute-value semantics hidden in reduced relations.

## 3. A DECOMPOSITION FRAMEWORK FOR ENTITY RELATED ATTRIBUTE-VALUE PAIRS

As illustrated in Figure 1, it includes three major steps: (1) pre-processing data; (2) acquiring reduced relations; and (3) training the CRF model and applying it to identify new attribute-value pairs. The CRF model is trained with the reduced relations labeled with semantic roles, such as $entity, class, attribute\ and\ value$.

### 3.1 Data Pre-processing

The first step in the proposed decomposition framework is to traverse over the text corpus for processing each sentence to detect reduced relations. We resort to linguistic processing techniques such as sentence boundary detection, part-of-speech (POS) tagging, phrase chunking and Named Entity Recognition.

Text in each document is first split into sentences using a sentence boundary detection tool[3]. A part-of-speech (POS) Tagger[4] is then used to annotate each sentence with POS tags. After that, we use a phrase chunking tool[5] to group word tokens into phrases. To detect as many potential entities as possible, a Name Entity Recognition (NER) tool[6] is first used and then the heuristic rules discussed in Section 3.2.1 are applied. Once we have completed processing the text corpus, we identify reduced relations as explained in Section 3.2.

### 3.2 Reduced Relation Acquisition

Ideally, the entity related information can be identified by searching for patterns in the text data [2, 11, 23]. However, linguistic patterns may easily become overfit and have difficulty to find quality information due to a large amount of noise present in the data. In this paper, we utilize the decomposed representation to address the quality issue. The entity related attribute-value pair is modeled as a triple of entity, attribute and value. The triple $r$ is then broken down into two tuples $r'$ and $r''$as shown in Figure 2.

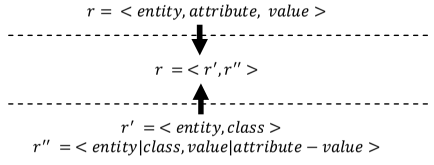



**Figure 2. Reduced relations for triple $r$.**

The "$value|attribute - value$" in $r''$indicates that it is possible that an entity can have implicit or explicit attribute for its value. These two tuples are modeled as $\{< arg1, rel, arg2 >\}$, where $args$ are any possible pair of left and right arguments of $r'$ or $r''$ , i.e, entity and class or entity and value, and $rel$ represents the textual fragment indicating semantic relation between two arguments. The reduced relation $r'$ is first detected from the context text. Once the relation between entity and its class is identified, the task is extended to identify the reduced relation $r''$ for finding attribute-value pairs. Next, we discuss how to find reduced relation $r'$ and $r''$.

#### *3.2.1 Syntactic dependency*

Syntactic dependency representation is designed to provide a description of the grammatical relations. It explains the relations between pairs of words in a sentence. For example, the Stanford parser dependencies of "Quebec City is the capital of the Canadian province of Quebec." are represented as in Figure 3a.

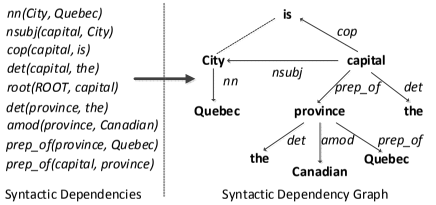



**Figure 3. Syntactic dependencies and corresponding syntactic dependency graph.**

Each dependency represents a relation between a pair of word tokens, e.g., "nn(City, Quebec)". We form these dependencies into a directed graph $G$ (Figure 3b), $G = < V, E >$, where $V$ is a set of nodes containing all word tokens in the sentence, e.g., "City" or "capital", and $E$ are edges denoting the relation between any pair of word tokens, e.g., "nsubj". We then use the shortest connecting path that includes subject, verb and object of a sentence to find relation:

$$City \xrightarrow{nsubj} capital \xleftarrow{cop} is$$

We call this as BasicRelation and it is useful to capture information of a basic relation. However, it loses semantic information on phrase level. For example, the BasicRelation does not indicate the integrity of entity "Quebec City". In order to capture meaningful relations, we expand the BasicRelation with phrase level information by adding modifier dependencies of the word tokens in BasicRelation, such as adverbial and adjectival modifiers as well as dependencies that modify verb, like "neg" and "aux". We utilize the expanded BasicRelations to derive our reduced relations $r'$ and $r''$.

#### *3.2.2 Finding Reduced Relation*

**Tuple r′.** We first examine the related entity with expanded BasicRelation to identify potential reduced relation $r'$. In $\{< arg1, rel, arg2 >\}$, if $arg1$ or $arg2$ contains the related entity, it is a potential reduced relation $r'$ . We then check if the relation, $rel$ indicates an IsA relation. The relation $rel$ is checked against with IsA relation pattern, IsA($arg1$, $arg2$), which can be summarized as:

(A1) [arg1] copula [arg2]

where *copula* represents any form of copula from the context where target entity appears. For example, the extended BasicRelation, "Quebec $City_{arg1}$ $is_{copula}$ $capital_{arg2}$" carries an IsA relation. If the $rel$ is not an IsA relation, the extended BasicRelation is then put into finding tuple $r''$.

**Tuple r″.** For $\{< arg1, rel, arg2 >\}$, if $arg1$ or $arg2$ contains related entity or extended class, it is considered as a reduced relation $r''$. For tuple $r''$, the $rel$ is not limited (except IsA

[3] http://nlp.stanford.edu/software/corenlp.shtml

[4] http://nlp.stanford.edu/software/tagger.shtml

[5] http://nlp.stanford.edu/software/corenlp.shtml

[6] http://nlp.stanford.edu/ner/index.shtml

relation) and contributes to the semantics of attribute-value pair. One example can be, "The city $_{arg1}$ is founded in $_{rel}$ $1834_{arg2}$". In the example, $arg1$ is an extended class and $rel$ reveals the semantics of attribute-value pair for the extended class.

The text segments, that contain the reduced relation $r'$ or $r''$, are added to the candidate pool of attribute-value pairs for semantic analysis as explained in next section.

## 3.3 Attribute-Value Pair Identification

We propose to use a discriminative model, Conditional Random Fields (CRF), for identification of attribute-value semantics in the attribute-value pair's candidate pool. We cast the problem of identifying attribute-value pairs as a joint segmentation or classification problem. Our goal is to semantically tag attribute and value for a related entity in reduced relations $r'$ and $r''$. The reduced relations are labeled with semantic roles such as $entity, class, attribute\ and\ value$.

### *3.3.1 Model*

**Conditional Random Fields (CRF).** The CRF model, a form of undirected graphical model, is a probabilistic framework for labeling sequential data [13]. Its definition is as follows:

*Given a graph $G = (V, E)$ where V denotes the nodes and E denotes the edges. Let $Y = (Y_v)_{v \in V}$ and $(X, Y)$ is a conditional random field conditioned on X when $Y_v$ obeys the Markov property with respect to G. X is a set of observed sequence input and Y is the set of random variables over the corresponding sequence. The probability of a set of labels Y for a sequence X under a linear chain CRF with features is:*

$$p(y|x) = \frac{1}{Z(x)} exp\Big(\sum_{v \in V,i} \lambda_i\, t_i\ (e, Y \mid e, X) + \sum_{v \in V,i} \mu_i\, s_i\ (v, Y \mid v, X)\Big) \quad (1)$$

Here $Z(x)$ is normalization factor, $s_i$ is a state feature function and $t_i$ is a transition feature function, $\lambda_i$ and $\mu_i$ are corresponding weights. The goal of using CRF is to obtain the marginal distribution of the labels $Y$ given an observed sequence $X$.

Let $X = (x_1, x_2, \dots, x_n)$ denote an input reduced relation $r'$ or $r''$ with word length of $n$. $Y = (y_1, y_2, \dots, y_k)$ represents the semantic labels of $k$ attribute-value pairs and $c$ is the class of the related entity $e$. Our goal is to obtain the most probable labelling sequence $Y$ of attribute-value pairs for an input X of text segment:

$$\hat{Y} = \underset{y}{argmax}\, p(y|c, e, x) \quad (2)$$

where the related entity $e$ and its class $c$ is identified in $r'$. Therefore, equation (2) can be written as:

$$\hat{Y} = \underset{y}{argmax}\, p(y|x) \quad (3)$$

Equation (3) is short for notional simplicity and denoting that the label and parameter space are entity- and class-independent.

### *3.3.2 Label Scheme*

In order to train the proposed CRF model, a label scheme is designed to tag reduced relations. We develop five types of label to tag each word, as shown in Table 1.

For attribute (A) and value (V), we further use character "Exp"

**Table 1. Label sets and their meaning for the proposed CRF model.**

| Label | Meaning |
|---|---|
| E | Related entity |
| C | Entity class |
| A | Entity attribute e.g., "population" |
| V | Value for corresponding attribute |
| O | Others that do not have above semantics |

and "Imp" for explicit value and implicit value, respectively. Explicit value means that the value is explicitly stated with its attribute and implicit value means that the value needs to be induced. A text segment may contain multiple words. We apply position labels to each word in the segment. Any text segment contains two positions: the beginning of the segment (B) and the rest of the segment (I). We assign "O" to words that do not contribute to any semantics. With these tags, reduced relations $r'$ or $r''$ have been tagged. For example,

***r′***. *Quebec(E-B) City(E-I) is(O) the(O) capital(C-B) of(O) the(O) Canadian(V-Imp-B) province(A-B) of(O) Quebec(V-Exp-B).*

***r″***. *The(E-B) city(E-I) is(O) founded(A-B) in(A-I) 1608(V-Exp-B).*

In the example of $r'$, "Canadian" is an implicit value for attribute "country", while explicit value "Quebec" has a corresponding attribute "province" in the text. For the example of $r''$, the $rel$, "founded in" serves as relation attribute and the $arg2$ "1608" is the explicit value.

### *3.3.3 Model Features*

In this section, we explore the integration of rich features, including not only transition features but also syntactic features and semantic features in the CRF model to identify attribute-value pairs.

**Transition Features.** A transition feature (*Trans*) indicates label transition between adjacent states in CRF. For example, in "Quebec(E-B) City(E-I)", the transition feature captures the label changing from $t_{j-1}$, "E-B" to $t_j$, "E-I". We only use the first-order transition feature.

**Syntactic Features.** The reduced relations, which have certain syntactic style, intend to have attribute-value pairs. We use word features, part-of-speech (POS) features and segment features as syntactic features.

A word feature (*W*) is a binary feature that indicates if a specific word co-occurs with a label. We generalize this feature to *n-grams* by applying a sliding window. Each word of the input sequence $w_{1:N}$ is sequentially viewed as the centre of a window with size $n$. In other words, a word feature inspects current position word as well as *n-grams* identity. In this way, the context word features are explored to consider long distance word dependency. Since a word feature follows the linear order principle, the corresponding POS tag of input word is considered as another syntactic feature. The pos feature (*Pos*) indicates whether a label occurs depending on the part-of-speech of the current word. The part-of-speech feature is also extended from the current word to its neighborhood with a size of *n*.

Based on POS tagging, words are organized into $k$ different segments by phrase chunking. These segments can provide a syntactic clue about that which words are in the same segment and which words are not. We refer to this feature as segment feature (*PC*). These segments are used to learn the co-occurrence

between labels and syntactic segments. In other words, a segment feature favors words appearing in the same or an adjacent segment. Furthermore, another type of segment feature (*RI*) is created by capturing segments in a reduced relation. The reduced relation has an inherent structure i.e., $\{< arg1, rel, arg2 >\}$, which we refer to as the self-supervised segment feature.

**Semantic Features.** Semantic features (*Sem*) concern what a word means and how it is related to attribute or value. We create semantic features based on Named Entity Recognition (NER) and semantic lexicons.

NER is implemented as a semantic feature to express what label a named entity class is related to. For example, if "Sydney Harbor" is labelled as value for attribute "located on", the CRF model captures the entity class "Location" as NER semantic feature. When a new named entity with same class "Location" occurs, it would be labeled as value for attribute "located on". The name entity classes used for NER include: Location, Person, Organization and Misc. We also extend the name entity class feature to neighborhood with the length of *n*.

Similar to the named entity class features, we create semantic lexicons to generalize semantic features. A lexicon is a list of words/phrases with same semantic meaning. For example, the attributes e.g., "country" or "population" can be grouped into an attribute lexicon. Similarly, a list of country names or state names, also form a value lexicon for corresponding attributes. Generally, the lexicon is built from a structured data table or domain ontological knowledge. However, this type of data source generally contains limited semantic information. In order to enrich semantic lexicons, we apply some heuristics:

$$(h_1) \qquad \underline{E}\ has|had\ \underline{Attr|Value}$$

The $\underline{E}$ represents an entity and $has|had$ implies the attribute and/or value. The heuristics in $h_1$ is applied on web-scale data and attributes and values discovered by $(h_1)$ are added to our semantic lexicons. If one semantic lexicon presents in the input data, the semantic lexicon feature will be activated and deactivated if not. To better incorporate semantic lexicons to the CRF model, we relax the exact matching to relatedness matching by measuring similarity between semantic lexicon elements and input data. In this paper, we adopt *Levenshtein distance* for the similarity function,

$$Sim(I_j, SL_i) = 1 - Lev_{distance}/|L| \qquad (4)$$

where $I_j$ and $SL_i$ represent the current word or segment of input data and $i$th element in the semantic lexicon. $|L|$ is the length of the $i$th element used to normalize *Levenshtein distance,* $Lev_{distance}$. The semantic element with max similarity score from equation (4) is then used as the semantic lexicon feature.

# 4. EVALUATION

## 4.1 Datasets

Two document collections are included in the experiment. One is the general purpose dataset available as Web documents to be used by a search engine. In this paper, we use Google and the dataset can be assumed as Web documents indexed by Google. In the experiment, we randomly select 30 seed entities for each entity class, *city* and *movie*. Each seed entity becomes a query (e.g., "Melbourne"). It is submitted to search engine for obtaining Web documents that contains the seed entity. Due to the fact that the higher rank a document appears in a search result, more relevance it has with the search query, the top $k$ documents ($k$ = 100) are collected. Some examples of seed entities are as follows:

$$Seed\ Entity_{city} = \{Melbourne, New\ York, Boston, Beijing, London, \dots\}$$

$$Seed\ Entity_{movie} = \{Transformers\ 3, Iceage\ 3, The\ hangover\ 2, \dots\}$$

Another dataset we used is Wikipedia. Similarly, using a seed entity to query Wikipedia documents, Wikipedia documents that contain seed entities are added to our experimental dataset. After the experimental dataset is pre-processed as discussed in Section 3.1, the proposed decomposition framework is applied to extract candidate sentences that contain seed entities. After removing duplicate and non-related sentences, 1000 reduced relations are collected for city domain and 1000 reduced relations for movie domain. All reduced relations are annotated using labels in Table 1 and split into 90/10 for training/testing of the CRF model.

## 4.2 Evaluation Metrics

Two evaluation metrics are used: Macro-average F1 (F1) and label accuracy (Acc). F1 is computed based on label precision and recall. More specifically, the precision ($P$) and recall ($R$), are calculated as the number of labels divided by the number of true positive labels and the number of correct labels divided by the number of true positive labels, respectively. The Macro-average F1-measure is then measured by precision and recall:

$$F1 = \frac{2 * P * R}{P + R}$$

Secondly, a label of a word is true positive if the label assigned by the trained CRF model matches with its correct label. Label accuracy is measured by the total number of labels divided by the total number of true positive predicted by CRF model.

## 4.3 Results and Discussion

### *4.3.1 Syntactic Features*

We organize different features into various feature sets to evaluate the performance of every single feature. The first experiment we did is to evaluate the performance of syntactic features. Although the single feature of POS or W does not perform well, the integration of POS and W feature provides a better average F1 score and label accuracy (54.4/66.9) than using any of these features alone as shown in Table 2. It implies that certain word with its syntactic clue is related to attribute-value pair. Both average F1 score and label accuracy in feature set (4) and (5), compared to feature set (3), obtain absolute gain when adding any segment feature (RI or PC). This indicates that attribute-value pairs co-occur with syntactic segments. Moreover, the PC feature offers a small gain than the reduced relation segment feature, RI. This may be caused by the nature of reduced relation that the relation part, $rel$, is a verb phrase, which generally includes a preposition word e.g., "in, of" as the end of the $rel$ part. This nature may damage the correct form of syntactic segments. Using all syntactic features and transition features (5) achieves the best performance.

### *4.3.2 Semantic Features*

In this section, we evaluate the performance of semantic feature and its integration with syntactic features. Results in (6) (7) (8) as shown in Table 2 prove the consistency of semantic feature boosting the performance of the CRF model for finding attribute-value pairs. This explains the dependency between attribute-value

**Table 2. Macro-average F1 (F1) and label accuracy (Acc) using CRF with different features.**

| | City | | Movie | | Average | |
|---|---|---|---|---|---|---|
| **Features (%)** | **F1** | **Acc** | **F1** | **Acc** | **F1** | **Acc** |
| **(1) Trans+W** | 48.1 | 61.2 | 50.5 | 62.0 | 49.3 | 61.6 |
| **(2) Trans+Pos** | 52.9 | 66.3 | 54.8 | 67.2 | 53.8 | 66.7 |
| **(3) Trans+W+Pos** | 53.7 | 66.4 | 55.3 | 67.4 | 54.4 | 66.9 |
| **(4) Trans+W+Pos+RI** | 56.0 | 68.7 | 59.7 | 70.5 | 57.8 | 69.6 |
| **(5) Trans+W+Pos+PC** | 61.8 | 71.1 | 64.2 | 73.7 | 63.0 | 72.4 |
| **(6) Trans+W+Pos+Sem** | 68.5 | 75.1 | 72.1 | 79.5 | 70.3 | 77.3 |
| **(7) Trans+W+Pos+PC+Sem** | 68.4 | 74.7 | 71.9 | 78.8 | 70.1 | 76.7 |
| **(8) Trans+W+Pos+RI+Sem** | 69.0 | 77.0 | 72.7 | 80.6 | 70.8 | 78.8 |
| **(9) Trans+W+Pos+PC+RI+Sem** | **69.4** | **77.1** | **73.5** | **80.9** | **71.4** | **79.0** |

pairs and knowledge domain. More specifically, it shows that an entity class contains different attributes and different entity classes implicitly select their inherent attributes and corresponding values. By integrating all syntactic and semantic features with transition features, we obtain the highest performance of the proposed CRF model.

### *4.3.3 Comparison with Baseline*

Previous approaches of identifying attributes and their values focused on pattern-based methods [2, 11, 23]. Generally, they apply a set of heuristic rules to fetch demanded pattern instances. Motivated by [12], we implement a pattern-based method as the baseline for comparison. All reduced relations are tagged with their part-of-speech (pos). We then apply a set of patterns on pos-tagged reduced relations to find entity level information. The pattern-based method consists of two steps, finding related entity with its attributes and then capturing the value for attributes. Two groups of patterns are created to extract binary relations, $rel(related\ entity, attributes)$ and $rel(attributes, value)$. We first discover related entity (E) with its attribute (ATTR) using patterns:

*(1)* E *with|without DT? RB? JJ?* ATTR
*(2)* *DT* ATTR *of DT? RB? JJ?* E
*(3)* *DT* E*'s RB? JJ?* ATTR
*(4)* E *has|had a|an RB? JJ?* ATTR

and then identify related attribute (ATTR) and value (V = {JJ | DT? RB? JJ? NN}) with patterns:

*(1)* ATTR *of DT? E is|was* V
*(2)* *DT? RB?* V ATTR
*(3)* *DT? JJ or* V ATTR
*(4)* *DT?* E*'s* ATTR *is|was* V
*(5)* *is|was|are|were* V *in|of* ATTR

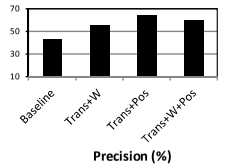
Figure 4 and Table 3 show the comparison between CRF model and the baseline on entity class, city. The result shows that our CRF model with only syntactic features outperforms the pattern-based method. The low recall and high precision of the pattern-based method indicates that pre-defined patterns face the overfitting problem. This also proves the effectiveness and robustness of the proposed decomposition framework and CRF model.

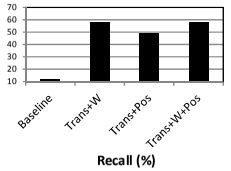
## 5. CONCLUSIONS

In this paper, we propose a novel decomposition framework integrating reduced relations and the discriminative model, CRF, for finding entity-related attribute-value pairs. In the decomposition framework, we first extract reduced relations and then automatically identify the semantics of attribute-value pairs using the model of CRF. Our experiment shows that the proposed CRF model achieves a high performance by integrating syntactic and semantic features with transition features. The proposed method outperforms the current state-of-art method based on patterns detection. In future, we would explore more semantic features, such as semantic lexicons describing the same reduced relation, and minimum supervision to train the CRF model.

**Table 3. Comparison between CRF and a pattern-based method.**

| | P | R | F1 |
|---|---|---|---|
| **Baseline (City) (%)** | 43.0 | 11.9 | 18.6 |
| **(1) Trans+Pos** | 55.3 | 57.9 | 52.9 |
| **(2) Trans+W** | 63.9 | 48.5 | 48.1 |
| **(3) Trans+W+Pos** | **60.1** | **58.1** | **53.6** |



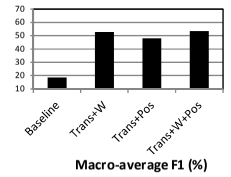



**Figure 4. Comparison between CRF models and a pattern-based method.**

## 6. REFERENCES


[1] Adafre, S. F., Rijke, de M., and Sang, E. T. K. 2007. Entity Retrieval. In Proceedings of International Conference of Recent Advances in Natural Language Processing (Borovets, Bulgaria, 2007). RANLP'07. John Benjamins, Amsterdam. Netherland.

[2] Almuhareb, A. 2006. Attributes in Lexical Acquisition. University of Essex, Colchester.

[3] Arguello, J., F. Diaz, F., Callan, J., and Crespo, J. F. 2009. Sources of evidence for vertical selection. In Proceedings of ACM International Conference on Research and development in information retrieval (Boston, MA, USA, 2009). SIGIR'09. ACM, New York, NY, 315–322. DOI= http://doi.acm.org/10.1145/1571941.1571997.

[4] Banko, M. 2009. Open Information Extraction for the Web. University of Washington, Seattle.

[5] Banko, M. and Etzioni, O. 2008. The Tradeoffs Between Open and Traditional Relation Extraction. In Proceedings of Annual Meeting of the Association for Computational Linguistics, (Ohio, USA, 2008). ACL'08. Association for Computational Linguistics, Stroudsburg, PA, 28–36.

[6] Bron, M., He, J., Hofmann, K., Meij, E., Rijke, M. D., Tsagkias, M., and Weerkamp, W. 2011. The University of Amsterdam at TREC 2010: Session, Entity and Relevance Feedback. In Proceedings of Text REtrieval Conference TREC 2010 (Gaithersburg, USA, 2011). TREC'11. NIST Special Publication, Gaithersburg, Maryland.

[7] Demartini, G., C. S. Firan, C. S., lofciu, T., Krestel, R., and Nejdl, W. 2010. Why finding entities in Wikipedia is difficult, sometimes. Inf. Retr, 135, 534–567. DOI= http://doi.acm.org/10.1007/s10791-010-9135-7.

[8] Etzioni, O., M. Banko, M., Soderland, S., and Weld, D. S. 2008. Open information extraction from the web. In Proceedings of International Joint Conference on Artificial Intelligence (Hyderabad, India, 2008). IJCAI'08. AAAI Press, Palo Alto, California, 2670–2676. DOI= http://doi.acm.org/10.1145/1409360.1409378 .

[9] Fader, A., Soderland, S., and Etzioni, O. 2011. Identifying relations for open information extraction. In Proceedings of Conference on Empirical Methods in Natural Language Processing (Edinburgh, United Kingdom, 2011). EMNLP'11. Association for Computational Linguistics, Stroudsburg, PA, 1535-1545.

[10] Ghani, R., K. Probst, K., Liu, Y., Krema, M., and Fano, A. 2006. Text mining for product attribute extraction. ACM SIGKDD Explorations Newsletter, 81, 41–48. DOI= http://doi.acm.org/10.1145/1147234.1147241 .

[11] Hartung, M. and Frank, A. 2010. A structured vector space model for hidden attribute meaning in adjective–noun phrases. In Proceedings of International Conference on Computational Linguistics (Beijing, China, 2010). COLING'10. Association for Computational Linguistics, Stroudsburg, PA, 430–438.

[12] Hartung, M. and Frank, A. 2011. Exploring supervised LDA models for assigning attributes to adjective–noun phrases. In Proceedings of Conference on Empirical Methods in Natural Language Processing (Edinburgh, United Kingdom, 2011). EMNLP'11. Association for Computational Linguistics, Stroudsburg, PA, 540–551.

[13] Lafferty, J. D., A. McCallum, A., and Pereira, F. C. N. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of International Conference on Machine Learning (Williamstown, USA, 2001). ICML'01. Morgan Kaufmann Publishers Inc, San Fransisco, CA, 282–289.

[14] Li, F., X. Zhang, X., Yuan, J.H., and Zhu, X.Y. 2008. Classifying what–type questions by head noun tagging. In Proceedings of International Conference on Computational Linguistics (Manchester, United Kingdom, 2008). COLING'08. Association for Computational Linguistics, Stroudsburg, PA, 481–488.

[15] Li, X. 2010. Understanding the semantic structure of noun phrase queries. In Proceedings of Annual Meeting of the Association for Computational Linguistics (Uppsala, Sweden, 2010). ACL'10. Association for Computational Linguistics, Stroudsburg, PA, 1337–1345.

[16] Pasca, M. 2007. Organizing and searching the world wide web of facts – step two: harnessing the wisdom of the crowds. In Proceedings of International conference on World Wide Web (Banff, Canada, 2007). WWW'07. ACM, New York, NY, 101–110. DOI= http://doi.acm.org/10.1145/1242572.1242587.

[17] Pasca, M. 2008. Turning web text and search queries into factual knowledge: hierarchical class attribute extraction. In Proceedings of National Conference on Artificial intelligence (Chicago, Illinois, 2008). AAAI'08. AAAI Press, Palo Alto, California, 1225–1230.

[18] Pasca, M. and Durme, B. V. 2007. What you seek is what you get: extraction of class attributes from query logs. In Proceedings of International joint conference on Artifical intelligence (Hyderabad, India, 2007). IJCAI'07. Morgan Kaufmann Publishers Inc, San Fransisco, CA, 2832–2837.

[19] Pasca, M. and Durme, B. V. 2008. Weakly–supervised acquisition of open–domain classes and class attributes from web documents and query logs. In Proceedings of Annual Meeting of the Association for Computational Linguistics (Ohio, USA, 2008). ACL'08. Association for Computational Linguistics, Stroudsburg, PA, 19–27.

[20] Reverb. http://reverb.cs.washington.edu

[21] Rode, H. 2008. From document to entity retrieval: improving precision and performance of focused text search. University of Twente, Enschede.

[22] Shen, D., J.–T. Sun, J.T., Yang, Q., and Chen, Z. 2006. Building bridges for web query classification. In Proceedings of ACM International Conference on Research and development in information retrieval (Seattle, USA, 2006). SIGIR'06. ACM, New York, NY, 131–138. DOI= http://doi.acm.org/10.1145/1148170.1148196.

[23] Sowa, John F. 2000. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Distributed Systems Online, 51, 1–3.

[24] Suchanek, F. M., Kasneci, G., and Weikum, G. 2007. YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. In Proceedings of International World Wide Web Conference (Banff, Canada, 2007). WWW'07. ACM,

New York, NY, 697–706. DOI= http://doi.acm.org/10.1145/1242572.1242667 .

[25] Tsikrika, T., P. Serdyukov, P., Rode, H., Westerveld, T., Aly, D, and Vries, A. P. 2008. Structured Document Retrieval, Multimedia Retrieval, and Entity Ranking Using PF/Tijah. In Proceedings of Focused access to XML documents: 6th international workshop of the initiative for the evaluation of XML (Dagstuhl Castle, Germany, 2008). INEX'08. Springer–Verlag, Heidelberg, Germany, 306–320. DOI= http://dx.doi.org/10.1007/978-3-540-85902-4_27.

[26] Voorhees, E. M. and Harman, D. 2004. Overview of the TREC 2004 Question Answering Track. In Proceedings of Text REtrieval Conference TREC–4 (Gaithersburg, USA, 2004). TREC'04. NIST Special Publication, Gaithersburg, Maryland, 1–11.

[27] Wu, F. and Weld, D. S. 2010. Open information extraction using Wikipedia. In Proceedings of Annual Meeting of the Association for Computational Linguistics (Uppsala, Sweden, 2010). ACL'10. Association for Computational Linguistics, Stroudsburg, PA, 118–127.

[28] Zirn, C., V. Nastase, V., and Strube, M. 2008. Distinguishing between instances and classes in the Wikipedia taxonomy. In Proceedings of European semantic web conference on The semantic web: research and applications (Tenerife, Spain, 2008). ESWC'08. Springer–Verlag, Heidelberg, Germany, 376–387. DOI= http://dx.doi.org/10.1007/978-3-540-68234-9_29.

# Is the Unigram Relevance Model Term Independent? Classifying Term Dependencies in Query Expansion

Mike Symonds[1], Peter Bruza[1], Guido Zuccon[2], Laurianne Sitbon[1], Ian Turner[3]

michael.symonds@qut.edu.au, p.bruza@qut.edu.au, guido.zuccon@csiro.au, l.sitbon@qut.edu.au, i.turner@qut.edu.au

[1] School of Information Systems, Queensland University of Technology
[2] Australian e-Health Research Centre, CSIRO
[3] School of Mathematical Sciences, Queensland University of Technology

Brisbane, Australia

## ABSTRACT

This paper develops a framework for classifying term dependencies in query expansion with respect to the role terms play in structural linguistic associations. The framework is used to classify and compare the query expansion terms produced by the unigram and positional relevance models. As the unigram relevance model does not explicitly model term dependencies in its estimation process it is often thought to ignore dependencies that exist between words in natural language.

The framework presented in this paper is underpinned by two types of linguistic association, namely syntagmatic and paradigmatic associations. It was found that syntagmatic associations were a more prevalent form of linguistic association used in query expansion. Paradoxically, it was the unigram model that exhibited this association more than the positional relevance model. This surprising finding has two potential implications for information retrieval models: (1) if linguistic associations underpin query expansion, then a probabilistic term dependence assumption based on position is inadequate for capturing them; (2) the unigram relevance model captures more term dependency information than its underlying theoretical model suggests, so its normative position as a baseline that ignores term dependencies should perhaps be reviewed.

## Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval Models

## General Terms

Algorithms, Experimentation, Theory

## Keywords

Query expansion, relevance models, linguistic associations


*ADCS '12* December 5-6, 2012, Dunedin, New Zealand.


## 1. INTRODUCTION

Within the information retrieval community it is understood that a user's query is often an imprecise description of their real information need. Therefore, there is a strong interest in the use of query expansion techniques. These techniques have been shown to provide significant improvements in retrieval effectiveness [1, 2, 3].

Early query expansion techniques did not explicitly use information about the dependencies that exist between terms in natural language [1, 9]. More recent approaches that explicitly model term dependencies have shown significant improvements in retrieval effectiveness over earlier techniques, and this improvement is often attributed to the *explicit* modelling of term dependencies [8, 3, 2]. For example, the unigram relevance model is often thought to ignore term dependencies. This assumption has been used to explain why dependency based approaches, like the positional relevance model, can significantly outperform the unigram relevance model.

In this paper we develop a novel framework to test this claim by comparing the extent to which linguistic associations are used within a unigram and positional relevance model. The framework is based on a recent query expansion approach, known as *tensor query expansion* (TQE), that uses features which have been shown to effectively measure the strength of linguistic associations.

A second contribution of this work is the discovery that although the unigram relevance model does not *explicitly* model term dependencies, unlike the positional relevance model, the estimation technique more effectively uses a form of term dependency underpinning most traditional dependency based query expansion approaches.

It is important to observe that this paper does not try to evaluate the effectiveness of each query expansion technique, but provides insight into the validity of claims relating to the cause of differences in retrieval effectiveness. Specifically those attributed to the explicit modelling of term dependencies within the query expansion process.

Section 2 introduces the relevance modelling framework, and outlines the unigram and positional relevance models. The novel framework for evaluating the types of linguistic associations within these query expansion techniques is presented in Section 3. This framework is applied in Section 4 to provide an empirical evaluation of the linguistic associations

modelled within the unigram and positional relevance models, before concluding remarks are presented in Section 5.

# 2. RELEVANCE MODELS

Most state-of-the-art document retrieval models are probabilistic in nature. These include the formally grounded family of language models [4]. In the language modelling framework, the query representations can be formally augmented using a relevance modelling approach [1]. This approach has been put forward as a strong benchmark in past information retrieval research investigating the effectiveness of query expansion approaches [3, 2].

## 2.1 Unigram relevance model

A relevance model estimates the probability of observing a word $w$ given some relevant evidence for a particular information need, represented by the query $Q$. The relevance model $P(w|R)$ is sampled from a multinomial distribution and is approximated as:

$$P(w|Q) = P(w|R) = \int_D P(w|D)P(D|Q)$$

$$\approx \frac{\sum_{D\in\mathcal{R}_Q} P(w|D)P(Q|D)P(D)}{\sum_w \sum_{D\in\mathcal{R}_Q} P(w|D)P(Q|D)P(D)}, \quad (1)$$

where $\mathcal{R}_Q$ is the set of documents pseudo-relevant or relevant to query $Q$, and $D$ is a document in $\mathcal{R}_Q$. To simplify the estimation, $P(D)$ is assumed uniform over this set of documents. In the unigram variant of the relevance model, where term dependencies are ignored, the estimate for $P(w|D)$ is often based on the Dirichlet smoothed term likelihood scores:

$$P(w|D) = \frac{df_w + \mu \frac{cf_w}{|C|}}{|D| + \mu}, \quad (2)$$

where $df_w$ is the document frequency of term $w$, $cf_w$ is the collection frequency of term $w$, $|C|$ is the word count in the collection, $|D|$ is the word count of the document and $\mu$ is the Dirichlet smoothing parameter. This form of the unigram relevance model will be referred to as **RM3** for the remainder of this paper.

## 2.2 Dependency based approaches

Dependency based query expansion approaches, such as the *positional relevance model* (PRM) [2] and *latent concept expansion* (LCE) [3], explicitly model term dependencies when producing expansion term estimates and this has been credited with producing significantly improved retrieval effectiveness over the unigram relevance model [2].

The LCE approach is a feature based approach set atop of the Markov random field (MRF) document ranking model. Since the MRF model uses term dependencies itself, and has been shown to be a stronger baseline than a unigram language model, it is not an appropriate comparison model for use in this investigation of the unigram relevance model. However, the PRM is set within the relevance modelling framework and hence uses the unigram language model for document ranking. This makes an appropriate comparison model for our investigation.

### 2.2.1 The Positional Relevance Model

Based on the intuition that topically related content is grouped together in text documents, the positional relevance model (PRM) uses proximity and positional information to produce expansion term estimates in the following way:

$$P(w|Q) = \frac{P(w,Q)}{P(Q)} \propto P(w,Q) = \sum_{D\in\mathcal{R}_Q} \sum_{i=1}^{|D|} P(w,Q,D,i), \quad (3)$$

where $i$ indicates a position in document $D$, and $\mathcal{R}_Q$ is the set of feedback documents (assumed to be relevant). Two sampling methods were proposed to estimate $P(w,Q,D,i)$. The first uses independent and identically distributed (*iid*) sampling, such that:

$$P(w,Q,D,i) \propto \frac{P(Q|D,i)P(w|D,i)}{|D|}. \quad (4)$$

The second approach to estimating $P(w,Q,D,i)$ uses conditional sampling, such that:

$$P(w,Q,D,i) = P(Q)P(D|Q)P(i|Q,D)P(w|D,i). \quad (5)$$

Both approaches are based on the following estimate:

$$P(w|D,i) = (1-\lambda)\frac{c'(w,i)}{\sqrt{2\pi\sigma^2}} + \lambda P(w|C) \quad (6)$$

where

$$c'(w,i) = \sum_{j=1}^{|D|} c(w,j) \exp\left[\frac{-(i-j)^2}{2\sigma^2}\right],$$

and $c(w,j)$ is the *actual* count of term $w$ at position $j$, $|D|$ is the length of the document, $\lambda$ is a smoothing parameter and $\sigma$ is used to parameterize the Guassian Kernel function.

The modelling of term dependencies within query expansion approaches are rarely linguistically motivated and often involve increasing the degrees of freedom of a model by adding free parameters, as seen in PRM. Being able to classify the types of linguistic associations modelled within a query expansion process could improve the understanding of the role term dependencies play in improving retrieval effectiveness. To this end, we aim to develop a framework for classify linguistic word associations used by query expansion techniques and test it by comparing the types of associations modelled within the RM3 and PRM approaches.

# 3. MODELLING WORD ASSOCIATIONS

To be able to classify the different types of linguistically meaningful word associations modelled within a query expansion technique, we use a recent corpus based model of word meaning, known as the *tensor encoding* (TE) model [5]. The TE model is grounded in structural linguistic theory, which states that that a meaning of a word is based on its relationships with other words. The two types of linguistic relationships underpinning meaning are: (i) syntagmatic and (ii) paradigmatic associations.

A *syntagmatic association* exists between two words if they co-occur more frequently than expected from chance. Some common examples may include "dog-bit" and "weather-rain". A *paradigmatic association* exists between two words if they can substitute for one another in a sentence without affecting the grammaticality or acceptability of the sentence. Some common examples may include "bit-chased" and "book-article".

## 3.1 The Tensor Encoding Model

The TE model provides a formal framework for combining measures of syntagmatic and paradigmatic associations that can be used to estimate the probability of observing a word $w$ given a priming word $q$, and can be stated as:

$$P(w|q) = \frac{1}{Z_\Gamma}\left[\gamma s_{\text{syn}}(q,w) + (1-\gamma)s_{\text{par}}(q,w)\right], \quad (7)$$

where $\gamma \in [0,1]$, mixes the amount of syntagmatic and paradigmatic features used in the estimation, and $Z_\Gamma = \sum_{w \in V_k}[\gamma s_{\text{syn}}(q,w) + (1-\gamma)s_{\text{par}}(q,w)]$, is used to normalise the distribution.

The TE model has been used to underpin an effective query expansion technique, known as *tensor query expansion* (TQE) [6]. For query expansion, the estimate in Equation (7) is extended to estimate the probability of observing a word $w$ given a sequence of query terms $Q = (q_1, \ldots, q_p)$, and can be expressed as:

$$P(w|Q) = \frac{1}{Z_\Gamma}\left[\gamma s_{\text{syn}}(Q,w) + (1-\gamma)s_{\text{par}}(Q,w)\right]. \quad (8)$$

To model syntagmatic and paradigmatic associations, the TE model is underpin by geometric representations of words that are automatically built from word order and co-occurrence information found in a set of training documents. The binding process that builds these geometric representations for each word involves moving a sliding context window across the text. The binding process for the second order TE model is defined as:

$$\boldsymbol{M}_w = \sum_{t \in CW}^{t \prec w}(r-d_t).\boldsymbol{e}_t \otimes \boldsymbol{e}_w^T + \sum_{t \in CW}^{t \succ w}(r-d_t).\boldsymbol{e}_w \otimes \boldsymbol{e}_t^T, \quad (9)$$

where $w$ is the focus term, $t$ is a non-stop word found within the sliding context window ($CW$), $k \prec w$ indicates that term $t$ appears before term $w$ in the context window, $k \succ w$ indicates that term $k$ appears after term $w$, $r$ is the radius of the sliding context window, and $d_k$ is the number of terms separating term $k$ and term $w$ within the context window. A context window is often referred to by its length. However, in the TE model the term *radius* is used to define the context window, as it better highlights the symmetric nature of the window and it also makes the equations behind the model less cumbersome in notation.

In a (pseudo) relevance feedback setting, the training documents, which the context window is slid across, refers to the top $k$ (pseudo) relevant documents. Once the text has been bound, each term is represented as a matrix in which the element values are proportional to the term-term co-occurrence frequencies. The generalised form of the matrix for term $w$ will be similar to:

$$\boldsymbol{M}_w = \begin{pmatrix} 0, & \ldots, 0, & f_{1w}, & 0, & \ldots, 0 \\ & & \ldots & & \\ 0, & \ldots, 0, & f_{(w-1)w}, & 0, & \ldots, 0 \\ f_{w1}, \ldots, & f_{w(w-1)}, & f_{ww}, & f_{w(w+1)}, & \ldots, f_{wN} \\ 0, & \ldots, 0, & f_{(w+1)w}, & 0, & \ldots, 0 \\ & & \ldots & & \\ 0, & \ldots, 0, & f_{Nw}, & 0, & \ldots, 0 \end{pmatrix},$$

where $f_{iw}$ is the value in row $i$ column $w$ of the matrix which represents the proximity scaled co-occurrence frequencies of term $i$ before term $w$, $f_{wj}$ is the value in row $w$ column $j$ of the matrix that represents the proximity scaled co-occurrence of term $j$ after term $w$, and $N$ is the number of unique terms in the vocabulary. This sparse representation is efficiently stored in low dimension storage vectors, that allow for computationally efficient similarity measures to be used on the terms.

Intuitively, in a (pseudo) relevance feedback setting strong syntagmatic associations between query terms and the other terms in the set of (pseudo) relevant documents are likely to exist. This is because most document ranking models, such as the unigram language model, score documents higher if they contain many instances of the query terms. Therefore, the top $k$ (pseudo) relevant documents will contain terms seen often around the query terms. This suggests that the expansion terms used to update the query representation within a (pseudo) relevance feedback setting, even those produced by the unigram relevance model will have strong syntagmatic associations with the query. To test this prediction, we can compare the sets of expansion terms produced by the unigram relevance model and syntagmatic measure of the TQE approach.

Within the TQE approach the strength of syntagmatic associations between a sequence of query terms $Q = (q_1, \ldots, q_p)$ and a vocabulary term $w$ can be measured using the cosine metric (i.e., the normalised dot product of the matrix representations), and simplifies to $s_{\text{syn}}(Q,w) =$

$$\frac{\sum_{\substack{j=1 \\ w \in Q}}^{N} s_w^2 f_{jw}^2 + \sum_{\substack{j=1 \\ j \neq w \\ w \in Q}}^{N} s_w^2 f_{wj}^2 + \sum_{\substack{i=q_1 \\ i \neq w}}^{q_m}(s_i^2 f_{iw}^2 + s_i^2 f_{wi}^2)}{\sqrt{\sum_{i=q_1}^{q_m}\left[\sum_{j=1}^{N} s_i^2 f_{ji}^2 + \sum_{\substack{j=1 \\ j \neq i}}^{N} s_i^2 f_{ij}^2\right]}\sqrt{\sum_{j=1}^{N} f_{jw}^2 + \sum_{\substack{j=1 \\ j \neq w}}^{N} f_{wj}^2}}, \quad (10)$$

where $q_1, \ldots, q_m$ are the unique query terms in $Q$ having $m \leq p$; $s_i$ is the number of times term $q_i$ appears in $Q$; $f_{ab}$ is the co-occurrence frequency of term $a$ appearing before term $b$ in the vocabulary; $f_{ba}$ is the co-occurrence frequency of term $a$ appearing after term $b$. This measure was shown to provide effective estimates for words most likely to succeed or precede another in text [5] and hence was reputed to be a reliable indicator of syntagmatic associations.

To complete the picture on how the other half of word meaning can be modelled within the TQE approach, a measure of the strength of *paradigmatic* associations between a sequence of query terms $Q = (q_1, \ldots, q_p)$ and a vocabulary term $w$, can be expressed as:

$$s_{\text{par}}(Q,w) = \frac{1}{Z_{\text{par}}}\sum_{j=q_1}^{q_p}\sum_{i=1}^{N}\frac{f_{\overline{ij}}.f_{\overline{iw}}}{\max(f_{\overline{ij}}, f_{\overline{iw}}, f_{\overline{wj}})^2}, \quad (11)$$

where $f_{\overline{ij}} = (f_{ji} + f_{ij})$, being the unordered co-occurrence frequency of terms $i$ and $j$; $N$ is the size of the vocabulary; max() returns the maximum argument value; and $Z_{\text{par}}$ normalizes the distribution. The use of the TE model's paradigmatic measure was shown to outperform human judgement and like models on a benchmark synonym judgement test [5].

Given the demonstrated effectiveness of these measures of syntagmatic and paradigmatic information, they will be used to underpin the framework developed in this paper for classifying linguistic associations modelled within the unigram and positional relevance models. Before applying this framework in an empirical evaluation, the similarities between the estimation techniques used by the syntagmatic feature in Equation (10) and unigram relevance model in Equation (2) will be demonstrated. This is to provide algebraic support to the intuition that syntagmatic associations are modelled within the unigram relevance model.

## 3.2 Use of Syntagmatic Associations

Research into the use of explicit term dependencies within the query expansion process found that when using information about syntagmatic associations a wider context window can lead to improved retrieval effectiveness [8]. That is, words far apart in a document can display strong syntagmatic associations.

We evaluated the retrieval effectiveness of the TQE approach, using solely the syntagmatic feature, $s_{\text{syn}}(.)$ to estimate query expansion terms within a pseudo relevance feedback setting on two TREC web data sets (Table 1). The results (Figure 1) indicate consistent improvements in retrieval effectiveness can be achieved by using larger context windows when modelling syntagmatic associations. Figure 1 also illustrates the robustness of $s_{\text{syn}}(.)$ for context window radii above 200. As the context window in the TE binding process does not cross document boundaries, it is worth considering the retrieval effectiveness achieved when the context window radius is set to each document's length. Using this radius, the MAP scores achieved by $s_{\text{syn}}(.)$ are 0.2491 and 0.0492 on the GOV2 (G2) and ClueWeb09 Category B (CW) data sets, respectively. This result indicates that superior retrieval effectiveness is achieved when syntagmatic associations between terms across the whole document are considered.

This condition can be modelled by the TE binding process in Equation (9) by setting the context window radius ($r$) equal to the document length ($|D|$). The resulting binding expression becomes:

$$\boldsymbol{M}_w = \sum_{t \in CW}^{t \prec w} (|D| - d_t).\boldsymbol{e}_t \otimes \boldsymbol{e}_w^T + \sum_{t \in CW}^{t \succ w} (|D| - d_t).\boldsymbol{e}_w \otimes \boldsymbol{e}_t^T. \tag{12}$$

The algebraic form of elements on row $w$ of the matrix $M_w$ in Equation (12) becomes:

$$f_{w,j} = \sum_{\substack{D \in \mathcal{R}_\mathcal{Q} \\ w \in D}} df_j(|D| - \bar{d}_{w,j}), \tag{13}$$

and on column $w$:

$$f_{i,w} = \sum_{\substack{D \in \mathcal{R}_\mathcal{Q} \\ w \in D}} df_i(|D| - \bar{d}_{i,w}), \tag{14}$$

where $D$ is a document in the set of pseudo relevant documents $\mathcal{R}_\mathcal{Q}$; $|D|$ is the length of document $D$; $df_j$ is the frequency of term $j$ in document $D$; $\bar{d}_{w,j}$ is the average number of terms separating term $w$ from term $j$ when $w$ is seen before $j$ in document $D$; and $\bar{d}_{i,w}$ is the average number of terms separating term $w$ from term $i$ when $w$ is seen after $i$ in document $D$.

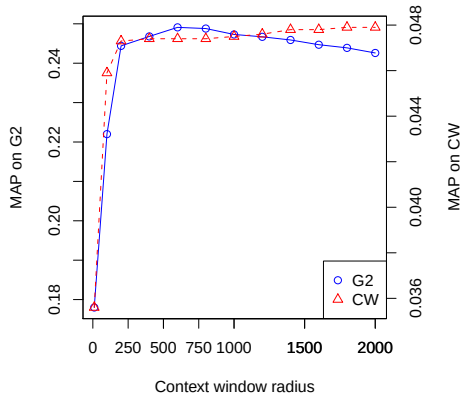



**Figure 1: Sensitivity of $s_{\text{syn}}(.)$ to window radius.**

| | Description | # Docs | Topics | $\overline{\lvert q \rvert}$ |
|---|---|---|---|---|
| G2 | 2004 crawl of .gov domain | 25,205,179 | 701-850 | 11 (4.1) |
| CW | Clueweb09 Category B | 50,220,423 | Web Track 51-150 | 9 (3.3) |

**Table 1: Overview of TREC collections and topic descriptions. $\overline{|q|}$ represents the average length of the queries, and the value in brackets is the standard deviation of the query lengths.**

When Equations (13) and (14) are substituted into Equation (10), the syntagmatic feature $s_{\text{syn}}(.)$ produces higher scores for terms that occur frequently (large $df_j$ and $df_i$) in the pseudo relevant documents. This result is similar to that produced by the Dirichlet smoothed likelihood estimation in Equation (1), which underpins RM3. However, Equation (1) contains a document normalisation factor. The cosine metric that defines $s_{\text{syn}}(.)$, also uses a form of normalisation that is linked to the document length. Equations (13) and (14) infer that terms that occur in larger documents will likely produce larger Frobenius norms (denominator of Equation (10)), and hence normalise the syntagmatic measure based on document length.

Therefore, the estimation techniques used in RM3 and $s_{\text{syn}}(.)$ (when the binding process in Equation 12 is used), are effectively based on term document frequencies and a document length normalisation factor. This result would lead us to believe that RM3 may be using very similar information to TQE's syntagmatic feature.

# 4. CLASSIFYING TERM DEPENDENCIES

The following section develops a framework to classify linguistic associations used within query expansion. Given that the TE model's syntagmatic feature has performed effectively on a word priming task and the paradigmatic feature has outperformed human judgement and like models on a benchmark synonym judgement task [5], we argue that they provide two reliable measures of structural linguistic associations.

The expansion terms used in the following analysis are produced during an ad hoc retrieval task carried out in a pseudo relevance feedback setting. Data set details are shown in Table 1. These TREC data sets are large web based collections that may make findings from these experiments relevant to web based applications. *Verbose* queries were chosen as they are long, discourse like queries, likely to provide sufficient term statistics to allow effective modelling of word associations within the TE model [7].

The experiments in this research were carried out using the Lemur Toolkit[1]. The Lemur implementation of the original positional relevance model is made available by the original authors[2]. The comparison of expansion terms is carried out using a Jaccard coefficient analysis and a Spearman's rank correlation coefficient analysis. The Jaccard coefficient analysis measures the average number of expansion terms that are common between two approaches. The Spearman's rank correlation coefficient is a finer grained analysis and measures, on a per query basis, how similar the overlap of

[1]The Lemur toolkit for language modelling and information retrieval: http://www.lemurproject.org
[2]http://sifaka.cs.uiuc.edu/~ylv2/pub/prm/prm.htm

| | | RM3 | PRM | $s_{\text{syn}}(.)$ |
|---|---|---|---|---|
| G2 | PRM | .509 (20) | 1 (30) | |
| | $s_{\text{syn}}(.)$ | .458 (19) | .362 (16) | 1 (30) |
| | $s_{\text{par}}(.)$ | .104 (6) | .108 (7) | .138 (6) |
| CW | PRM | .634 (23) | 1 (30) | |
| | $s_{\text{syn}}(.)$ | .466 (19) | .437 (18) | 1 (30) |
| | $s_{\text{par}}(.)$ | .131 (7) | .130 (7) | .144 (8) |

**Table 2: Average Jaccard co-efficients for the sets of expansion terms produced on the G2 and CW data sets for the best performing RM3, PRM, TQE syntagmatic and paradigmatic features. The average number of expansion terms that overlap between approaches is shown in brackets.**

two sets of expansion terms are with a third set.

The models used in the evaluation include RM3 and PRM. PRM was included in the evaluation to provide a benchmark for the amount of linguistic information being used by a technique that *explicitly* models term dependencies.

Given the focus is on comparing the expansion terms produced by each estimation technique, all common model parameters were fixed, including the number of feedback documents (30) and the number of expansion terms (30).

For each of the query expansion techniques, the remaining free model parameters were trained using 3-fold cross validation with the objective function maximising the MAP metric. This includes training the $\mu$ in Equation (2) for RM3. The free parameters trained for PRM include both $\sigma$ and $\lambda$ in Equation (6). The baseline unigram language model, used as the document scoring technique for all approaches, was run using the Lemur default parameters. The syntagmatic and paradigmatic features were built on a semantic space using a context window radius of 200 and 1 respectively.

Table 2 reports the Jaccard coefficients for the sets of expansion terms produced by RM3, PRM and the TQE syntagmatic and paradigmatic features. When compared to RM3, the syntagmatic feature $s_{\text{syn}}(.)$ has a minimum Jaccard coefficient of 0.458 (Table 2). This means that on average at least 19 out of 30 expansion terms suggested by $s_{\text{syn}}(.)$ are in common with those suggested by RM3.

As a comparison, PRM has a minimum Jaccard coefficient of 0.362 with $s_{\text{syn}}(.)$. This implies that on average at least 16 of 30 expansion terms are in common between PRM and $s_{\text{syn}}(.)$. This suggests that both RM3 and PRM use syntagmatic information when estimating query expansion terms, and that in fact RM3 has a stronger claim to the use of this form of term dependency.

Table 2 also shows that on average RM3 and PRM share a maximum of 7 expansion terms (out of 30) with those produced by TQE's paradigmatic measure $s_{\text{par}}(.)$. This result suggests that both RM3 and PRM use very little information about paradigmatic associations in their estimation process.

To investigate the overlap for each topic, a per-topic Spearman's rank correlation coefficient analysis, along the number of overlapping expansion terms on the $s_{\text{par}}(.)$ feature, was performed for the RM3, PRM and $s_{\text{syn}}(.)$ approaches. The resulting coefficients were, $\rho_{(PAR:SYN,RM3)} = 0.941$, $\rho_{(PAR:SYN,PRM)} = 0.863$ and $\rho_{(PAR:RM3,PRM)} = 0.883$. This result again suggests that RM3 may be using more information about syntagmatic associations than PRM.

The above discussion provides empirical and theoretical evidence to suggest that in augmenting the query model, RM3 uses information about syntagmatic associations. Given the linguistic credentials of TE model's syntagmatic feature, this research raises questions over the claim that dependency based approaches, like PRM and LCE, significantly outperform RM3 due to their use of explicit modelling of term dependencies. The gap in retrieval effectiveness may then be due to other factors.

# 5. CONCLUSION

The framework outlined in this paper provides a valuable method for classifying linguistic associations used within query expansion. We believe this framework can help information retrieval researchers better understand the types of linguistic term dependencies that may be responsible for differences in retrieval effectiveness.

This was demonstrated by using the framework to compare the strength of syntagmatic and paradigmatic associations displayed in query expansion terms for the unigram and positional relevance models. We found that not only do the best expanded query models for each approach display heavy use of syntagmatic associations, but the unigram relevance model has a stronger reliance on these syntagmatic associations. This leads us to question the claim that the unigram relevance model is outperformed by dependency based query expansion approaches because they use term dependencies.

# 6. REFERENCES


[1] V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR '01*, pages 120–127, New York, NY, USA, 2001. ACM.

[2] Y. Lv and C. Zhai. Positional relevance model for pseudo-relevance feedback. In *SIGIR '10*, SIGIR '10, pages 579–586, New York, NY, USA, 2010. ACM.

[3] D. Metzler and W. B. Croft. Latent concept expansion using markov random fields. In *SIGIR '07*, pages 311–318, New York, NY, USA, 2007. ACM.

[4] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR '98*, pages 275–281, New York, NY, USA, 1998. ACM.

[5] M. Symonds, P. Bruza, L. Sitbon, and I. Turner. Modelling word meaning using efficient tensor representations. In *PACLIC '11*, pages 313–322, 2011.

[6] M. Symonds, P. Bruza, L. Sitbon, and I. Turner. Tensor Query Expansion: a cognitive based relevance model. In *Australasian Document Computing Symposium 2011*, pages 87–94, 2011.

[7] M. Symonds, G. Zuccon, B. Koopman, P. Bruza, and A. Nguyen. Semantic judgement of medical concepts: Combining syntagmatic and paradigmatic information with the tensor encoding model. In *ALTA '12*, pages 87–94, 2012.

[8] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *SIGIR '96*, pages 4–11, New York, NY, USA, 1996. ACM.

[9] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01*, pages 403–410, New York, NY, USA, 2001. ACM.

# Pairwise Similarity of TopSig Document Signatures

Christopher M. De Vries
Electrical Engineering and Computer Science
Queensland University of Technology
Brisbane, Australia
chris@de-vries.id.au

Shlomo Geva
Electrical Engineering and Computer Science
Queensland University of Technology
Brisbane, Australia
s.geva@qut.edu.au

## ABSTRACT

This paper analyses the pairwise distances of signatures produced by the TopSig retrieval model on two document collections. The distribution of the distances are compared to purely random signatures. It explains why TopSig is only competitive with state of the art retrieval models at early precision. Only the local neighbourhood of the signatures is interpretable. We suggest this is a common property of vector space models.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval Models*

## Keywords

Signature Files, Topology, Vector Space IR, Random Indexing, Document Signatures, Search Engines, Document Clustering, Near Duplicate Detection, Relevance Feedback

## 1. INTRODUCTION

This paper investigates the properties of the pairwise similarities of document signatures produced by TopSig. TopSig is a retrieval model where documents are represented by $d$-bit binary strings that lie on a $d$-dimensional collection hypercube. The signatures are produced by a random process called random indexing [10] or random projection [1] which compresses the standard term-by-document matrix.

Pairwise similarity plays an important role in many information retrieval related tasks such as ad hoc retrieval, clustering, classification, filtering, near duplicate detection and relevance feedback.

The paper proceeds as follows. In Section 2, the TopSig retrieval model is introduced. Section 3 describes the document collections used in the experiments. The experimental setup is introduced in Section 4 and the results are presented in Section 5. The paper is concluded by a discussion of the implications of the results in Section 6.


*ADCS '12* December 5-6, 2012, Dunedin, New Zealand.


## 2. TOPSIG

TopSig [7] offers a radically different approach to the construction of file signatures. Traditional file signatures [6] have been shown to be inferior to approaches using inverted indexes, both in terms of the time and space required to process and store the index [12, 13]. However, TopSig overcomes previous criticisms aimed at file signatures by taking a principled approach using the vector space model, dimensionality reduction and numeric quantisation. Previous approaches to file signatures were constructed in an ad hoc fashion by combining random binary signatures using a bitwise XOR which is a Bloom filter [2] for the terms contained in documents. In contrast, TopSig randomly indexes a weighted term-by-document matrix and then quantises it. TopSig is competitive with state of the art probabilistic and language retrieval models at early precision, and clustering approaches [7].

Let $D = \{d_1, d_2, ..., d_n\}$ be a document collection of $n$ documents signatures, $D \subset \{+1, -1\}^d$, $|D| = n$. Let $F = \{f_1, f_2, ..., f_n\}$ be the same document collection as $D$ where each document is represented by a $v$-dimensional real valued vector, $F \subset \mathbb{R}^v$, $|F| = n$, where $v$ is the size of the vocabulary of the document collection. $F$ is the term-by-document matrix in the full space of the collection vocabulary which underlies most modern retrieval systems.

TopSig indexes documents using a mapping function, $m : \mathbb{R}^v \rightarrow \{+1, -1\}^d$, that maps a document from the original $v$-dimensional continuous real valued term space, to a $d$-dimensional discrete binary valued space. The index is constructed using a mapping function, $D = \{f \in F : m(f)\}$. The mapping function creates a sparse random ternary index vector of $d$-dimensions for each term in the document with +1 and -1 values in random positions and the majority of positions containing 0 values. These randomly generated codes are almost orthogonal to each other and have been shown to provide comparable quality to orthogonal approaches such as principle component analysis [1]. The index vector is multiplied by the term weight and added to a $d$-dimensional real valued vector that represents the document. Once all the terms in a document have been processed, this reduced dimensionality document vector is then quantised to a $d$-dimensional binary vector by thresholding each value in each dimension to 1 if greater than 0 and 0 otherwise. The 1 and 0 values in the binary vector represent +1 and -1 values. This mapping function can be applied to each document independently, meaning that new documents can be indexed in isolation without having to update the existing index. This is a key advantage to random indexing [10]

over other dimensionality reduction techniques such as latent semantic analysis [5] which requires global analysis of the term-by-document matrix using the singular value decomposition [8].

The indexing process of TopSig is similar to that of SimHash [3]. However, TopSig uses signatures an order of magnitude longer than SimHash and it uses much sparser random codes. The search process for ad hoc retrieval also differs, where TopSig searches in the subspace of the query and applies relevance feedback.

The binary vectors in $D$ provide a faithful representation of the original document vectors in $F$. The topological relationships in the original space are preserved in the reduced dimensionality space. This is supported by the Johnson-Lindenstrauss lemma [9] that states if points in a high-dimensional space are projected into a randomly chosen subspace, of sufficiently high-dimensionality, then the distances between the points are approximately preserved. It also states that the number of dimensions required to reproduce the topology is asymptotically logarithmic in the number of points.

## 3. DOCUMENT COLLECTIONS

We have used the INEX Wikipedia 2009 collection and the TREC Wall Street Journal (WSJ) Collection to evaluate pairwise distances of TopSig signatures. The INEX Wikipedia collection contains 2,666,190 documents with a vocabulary of 2,132,352 terms. We have used 2 subsets of this collection during evaluation. The first is a 144,265 document subset used for the INEX 2010 XML Mining track [4]. This is the reference run for the ad hoc track in 2010 produced by an implementation of Okapi BM25 in the ATIRE search engine [11]. It is denoted by $\text{INEX}_{reference}$. The second is a randomly selected 144,265 document subset chosen to match the size of the XML Mining subset. It is denoted by $\text{INEX}_{random}$. Subsets of the INEX Wikipedia 2009 collection were used for this experiment because calculating pairwise distances has a time complexity of $O(n^2)$ and becomes intractable for millions of documents. The mean document length in the Wikipedia has 360 terms, the shortest has 1 term and the longest has 38,740 terms. The Wall Street Journal Collection consists of 173,252 documents and a vocabulary of 113,288 terms. The mean WSJ document length is 475 terms, the shortest has 3 terms, and the longest has 12,811 terms.

The INEX Wikipedia 2009 collection consists of 12GB of uncompressed text or 50GB of uncompressed XML which includes semantic markup. The 2,666,190 documents are split into 3,617,380 passages. 1024-bit TopSig signatures use a total of 441MB to index the collection. The TREC Wall Street Journal consists of 518MB of uncompressed text. The 173,252 documents are split into 222,238 passages. 1024-bit TopSig signatures use a total of 27MB to index the collection.

## 4. EXPERIMENTAL SETUP

Pairwise similarities define the topology of a set of documents. Each document is compared to every other document. These similarities define the relationships between all documents in a collection. If two documents are nearby each other they share the same semantic context. TopSig uses the Hamming distance to measure similarity between two documents. It produces values in the range $[0, d]$ where 0 indicates the documents are indentical and values from 1 to $d$ indicate decreasing similarity between documents where $d$ is the most dissimilar two documents can be.

The TopSig indexing process uses random codes to compress document vectors. These random codes are also called index vectors in the random indexing process. The codes are influenced by the original document vectors. Similar documents are placed close together in the reduced binary vector space that are close together in the original vector space. Therefore, it is expected that the pairwise relationships between documents will be biased by this process. If the indexing process has no effect then the document signatures would appear no different to purely random signatures. The pairwise distances between randomly generated random signatures can be described by the Binomial distribution. The distribution of pairwise distances produced by the TopSig indexing process can be estimated by creating a histogram of similarity counts at all Hamming distances.

All $\frac{n^2-n}{2}$ pairwise distances between document signatures in $D$ are calculated. This is all the similarities contained in the upper triangular form of the pairwise distance matrix without the entries along main diagonal. The lower half of the pairwise distance matrix does not need to be calculated as the Hamming distance is symmetric. Measuring a pair of signatures both ways around does not add any extra information. The Hamming distance similarity function, $s : \{+1, -1\}^d \times \{+1, -1\}^d \rightarrow \mathbb{N}$, is symmetric such that two documents compared in either order produce the same result, $d_x, d_y \in D : s(d_x, d_y) = s(d_y, d_x)$. The estimated probability of finding a signature at Hamming distance, $h$, is the fraction of similarities at that distance over the total number of distance comparisons. The probability mass function, $pmf_e : \mathbb{P}\{+1, -1\}^d \times \mathbb{N} \rightarrow \mathbb{R}$, produces the estimated probability from the pairwise distances in $D$ where $n$ is the number of signatures in the collection $D$, $|D| = n$,

$$pmf_e(D, h) = \frac{\left|\left\{(d_x, d_y) : \begin{array}{c} d_x, d_y \in D \wedge d_x \neq d_y \\ \wedge s(d_x, d_y) = h \end{array}\right\}\right|}{\frac{n^2-n}{2}}. \quad (1)$$

Note that $pmf_e$ is the estimated probability for finding a signature at distance, $h$, when averaged across all documents in the collection, $D$.

The probability of finding a random binary code of length, $d$, at Hamming distance, $h$, is described by the Binomial probability mass function, $pmf_b : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R}$,

$$pmf_b(d, h) = \left( \begin{array}{c} d \\ h \end{array} \right) p^h (1-p)^{d-h}, \quad (2)$$

where $p$ is the probability of a bit being set, $p = 0.5$.

The cumulative distribution function for either the estimated, $cdf_e : \mathbb{P}\{+1, -1\}^d \times \mathbb{N} \rightarrow \mathbb{R}$, or Binomial, $cdf_b : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R}$, probability distributions are the sum of the probability mass function from 0 to $h$,

$$cdf_e(D, h) = \sum_0^h pmf_e(D, h), \quad (3)$$

$$cdf_b(d, h) = \sum_0^h pmf_b(d, h). \quad (4)$$

| Collection | Documents | Signatures (Passages) |
|---|---|---|
| $\text{INEX}_{reference}$ | 144,265 | 328,207 |
| $\text{INEX}_{random}$ | 144,265 | 195,369 |
| WSJ | 173,252 | 222,238 |

**Table 1: Number of Signatures Generated by TopSig**

An implementation of the TopSig [1] search engine was used to index the document collections. It splits documents into passages on a sentence boundary between a minimum and maximum number of word tokens. If the maximum word token limit is reached before the end of a sentence, it is split at that point. Therefore, documents have multiple signatures. This has been found to be effective for retrieval of documents of varying length. The INEX collection was split on a minimum of 256 and maximum of 280 word tokens. The WSJ collection was split on a minimum of 256 and a maximum of 384 word tokens. All indexes use 1024-bit signatures, resulting in the number of signatures as listed in Table 1.

The resulting probability distributions have been multiplied by the number of signatures in a collection to produce the expected number of signatures at a given Hamming distance. In this case, the *pmf* gives the average number of signatures expected at a particular Hamming distance when comparing a signature to the entire collection. The *cdf* gives the average number of signatures expected to lie within a given Hamming distance when comparing a signature to the entire collection, i.e. the number of nearest neighbours to expect within a particular Hamming distance.

## 5. EXPERIMENTAL RESULTS

Figures 1 to 12 highlight the difference between the distributions estimated from the pairwise distances and the distributions expected from random binary signatures from the Binomial distribution. It can be seen that all the estimated distributions are left skewed towards a Hamming distance of 0. This indicates that the signatures produced by TopSig are biased in such a way that documents are more similar to each other. There are more documents expected at a more similar, lower Hamming distance, than expected at random.

The probability mass functions in Figures 1, 2 and 3 represent the expected number of signatures to be seen at a particular Hamming distance. The graphs have been centred around the middle of the distributions to allow better visualisation of the separation between the distributions. The tails of the distributions tend towards 0 as expected. For example, the graph in Figure 3 has a y value for the estimated distribition of 1033.39 at a Hamming distance of 441. When comparing a signature to the entire collection, it would be expected on average to encounter 1033.39 signatures that are exactly at a Hamming distance of 441. However, the expected number of signatures at a Hamming distance of 441 for purely random signatures is only 0.29. This suggests that the signatures produced by TopSig are not uniformly distributed throughout the feature space. The number of nearest neighbours at a given Hamming distance, as described by the *pmf*, quickly increases when starting from a Hamming

[1] `http://topsig.googlecode.com`

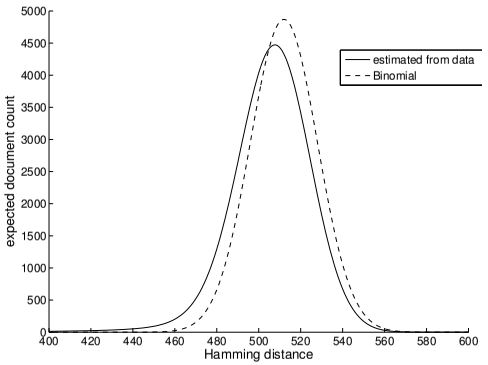



**Figure 1: $\text{INEX}_{random}$ *pmf***

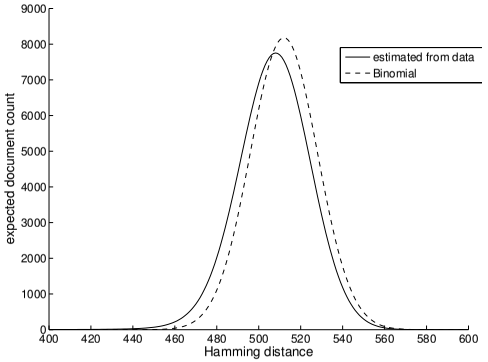



**Figure 2: $\text{INEX}_{reference}$ *pmf***

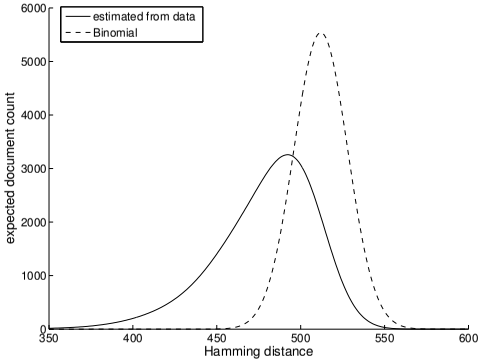



**Figure 3: WSJ *pmf***

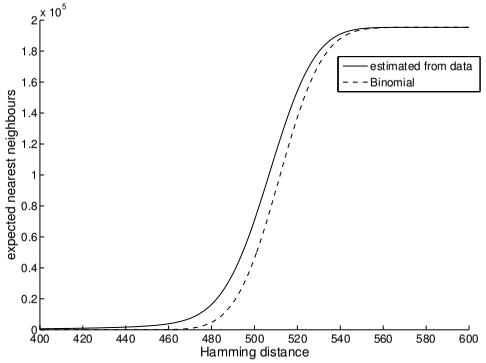



**Figure 4: INEX$_{random}$ *cdf***

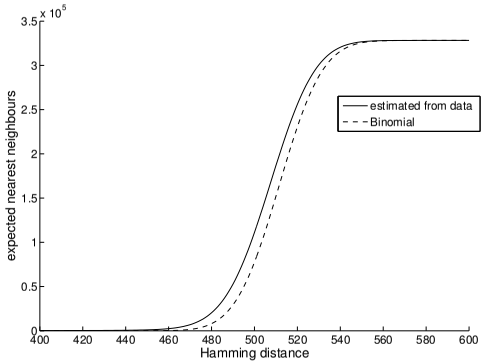



**Figure 5: INEX$_{reference}$ *cdf***

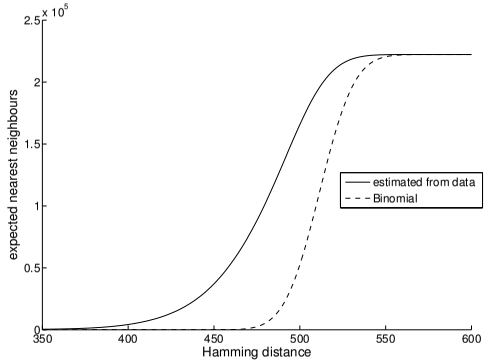



**Figure 6: WSJ *cdf***

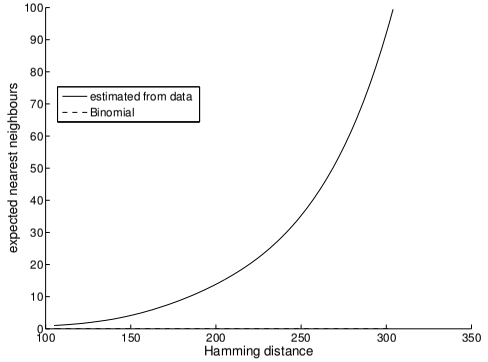



**Figure 7: INEX$_{random}$ *cdf***
**First 100 signatures from estimated distribution**

distance of 0 and proceeding to a Hamming distance of $d$. This is the same order that TopSig ranks signatures in the ranked list, or, any other task that compares relative orderings of documents such as clustering. This is true for both the estimated and Binomial distributions. As the neighbourhood of analysis is increased, more and more documents become equidistant; i.e. they share the same Hamming distance. This is a property of vector space models known as the "curse of dimensionality". However, the left skewness of estimated distributions indicates that the pairwise distances of the document collections allow better differentiation between documents than expected at random. It is this left skewness of the distributions that allows TopSig to compete with state of the art retrieval models at early precision. Documents are topically clustered and are not random bags of words. Neither of the document collections have signatures further apart than a Hamming distance of 617, meaning that the indexing process has moved the random signatures from the right side of the distribution to the left. This again indicates that similar signatures are being placed closer together and are therefore more topically related and clustered.

The cumulative distribution functions in Figures 4, 5 and 6 represent the area under the curve for each of the probability mass functions. The y value at a given Hamming distance indicates the average number of nearest neighbours expected within a given Hamming distance when comparing a signature to the entire collection. For example, the graph in Figure 6 has a y value for the estimated distribition of 25975.78 at a Hamming distance of 441. When comparing a signature to the entire collection, it would be expected on average to encounter 25975.78 signatures that are nearest neighbours at a Hamming distance of 441. However, the expected number of signatures at a Hamming distance of 441 for purely random signatures is only 1.13. Again, the separation between the curves indicates that TopSig is placing semantically related documents close together and preserving the topological relationships of the original document vectors.

Figures 7, 8 and 9 zoom in on the *cdf* where the first 100 nearest neighbours are expected for the distribution estimated from the pairwise distances of the collections. In all cases almost zero signatures are expected at random where

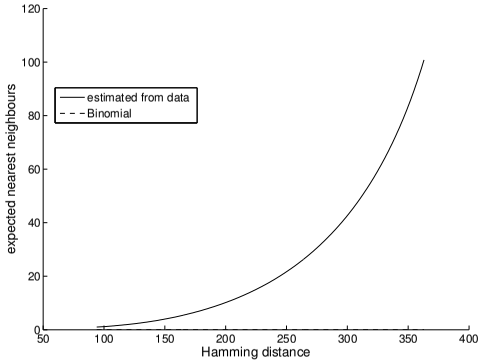



**Figure 8: INEX$_{reference}$ *cdf***
**First 100 signatures from estimated distribution**

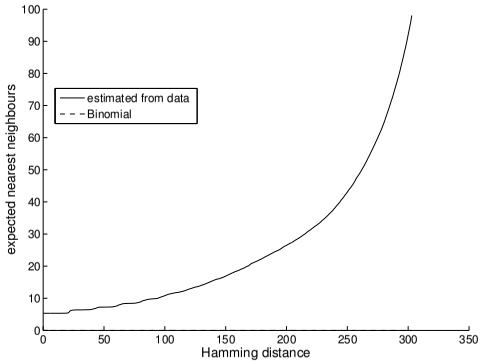



**Figure 9: WSJ *cdf***
**First 100 signatures from estimated distribution**

| Collection | $cdf_b$ @ $cdf_e = 1$ | $cdf_b$ @ $cdf_e = 100$ |
|---|---|---|
| INEX$_{reference}$ | $1.66 \times 10^{-168}$ | $1.104 \times 10^{-15}$ |
| INEX$_{random}$ | $1.80 \times 10^{-158}$ | $2.06 \times 10^{-34}$ |
| WSJ | 0 | $1.59 \times 10^{-34}$ |

**Table 2: Nearest Neighbours Expected from $cdf_b$**

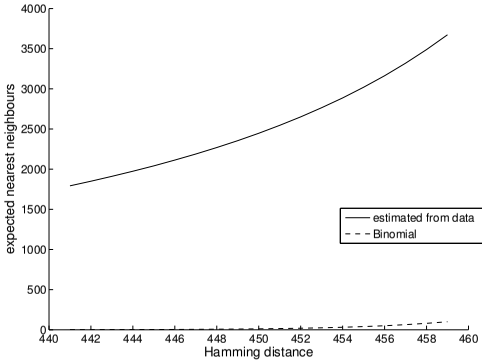



**Figure 10: INEX$_{random}$ *cdf***
**First 100 signatures from Binomial distribution**

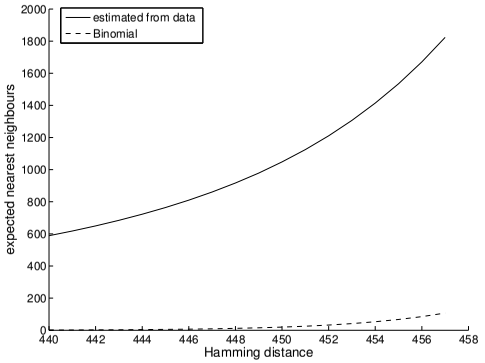



**Figure 11: INEX$_{reference}$ *cdf***
**First 100 signatures from Binomial distribution**

there are TopSig signatures expected in the range $[1, 100]$. This indicates that the signatures produced by TopSig return nearest neighbours at a Hamming distance much earlier than expected by purely random signatures. The start and end points of these curves are listed in Table 2.

Figures 10, 11 and 12 zoom in on the *cdf* where the first 100 nearest neighbours are expected for random binary signatures as described by the Binomial distribution. The start and end points of these curves are listed in Table 3. There are many more signatures expected to be nearest neighbours when using TopSig signatures. However, both the estimated and Binomial distributions have many equidistant documents around the middle of their distributions. This suggests that only the local neighbourhood of the signatures has semantic meaning. This can also be seen in the *pmf* distributions where most of the signatures exist around the middle of the distribution. Another perspective is that are too many ties at these distances for the feature space to differentiate signatures.

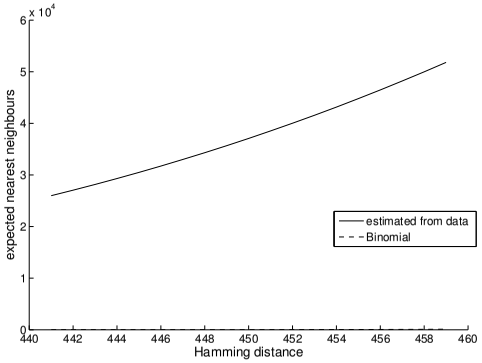



**Figure 12: WSJ *cdf***
**First 100 signatures from Binomial distribution**

| Collection | $cdf_e$ @ $cdf_b = 1$ | $cdf_e$ @ $cdf_b = 100$ |
|---|---|---|
| $\text{INEX}_{reference}$ | 567.86 | 2129.02 |
| $\text{INEX}_{random}$ | 1793.26 | 3673.78 |
| WSJ | 25495.67 | 50709.94 |

**Table 3: Nearest Neighbours Expected from $cdf_e$**

The skewness of the estimated distributions suggests that the feature space is not uniform and is clustered. Some areas of the space are more dense than others. This is vital for any document representation because it is this non-uniformity that allows differentiation of meaning.

Table 3 lists the number of nearest neighbours expected from the distribution estimated from pairwise distances when the Binomial distribution expects 1 and 100 nearest neighbours, as listed in columns 2 and 3 respectively. For example, the $\text{INEX}_{random}$ collection expects on average 1793.26 signatures to be nearest neighbours to other signatures when purely random signatures would expect 1. When purely random signatures expect on average 100 nearest neighbours, the $\text{INEX}_{random}$ collection expects 3673.78 nearest neighbours. These values are linearly interpolated as they exist in between two Hamming distances under the *cdf*. These values are the start and end points for the curves in Figures 10, 11 and 12. Table 2 lists the opposite, i.e., the number of nearest neighbours expected from the Binomial distribution when the distribution estimated from pairwise distances expects at 1 and 100 nearest neighbours.

| Collection | $\frac{cdf_b@h=512}{n}$ | $\frac{cdf_e@h=512}{n}$ |
|---|---|---|
| $\text{INEX}_{reference}$ | 0.51 | 0.61 |
| $\text{INEX}_{random}$ | 0.51 | 0.63 |
| WSJ | 0.51 | 0.89 |

**Table 4: Signatures Expected within $\frac{d}{2}$**

Table 4 lists the number of signatures expected within a Hamming distance of $\frac{d}{2}$. This summarises the distributions in a single number, where the difference between the distributions indicates the fraction of the signatures shifted from the left hand side of the Binomial distribution to the right by the indexing process. It is also the value under the *pmf* at $\frac{d}{2}$ which is also the y value of the *cdf* at $\frac{d}{2}$.

The difference in distributions between the INEX reference and random subsets indicates that the reference run is not suitable for estimating properties of the entire collection. This is to be expected as the reference run has been biased by the queries used for ad hoc retrieval. Table 3 shows that $\text{INEX}_{reference}$ expects 567.86 nearest neighbours where as $\text{INEX}_{random}$ expects 1793.26 nearest neighbours when purely random signatures expect 1 nearest neighbour. This indicates that the reference run is less clustered than than a random sample from the INEX Wikipedia collection. This can explained because the documents returned by the reference run are more diverse than a random sample from the collection. As the diverse topics are further apart, i.e. more dissimilar, there are more inter-topic distances than intra-topic distances, leading to less signatures being located nearby. Note that the reference run is determined by only searching in a few dimensions determined by the keywords in the queries, where as the pairwise distances compare entire documents, using their entire vocabulary.

# 6. DISCUSSION

The results presented indicate why TopSig is only competitive at early precision in comparison to probabilistic and language models for ad hoc retrieval. As the Hamming distance increases when proceeding down the ranked list more and more documents become equidistant. This can be seen in Figures 1, 2 and 3 containing plots of probability mass functions indicating the expected number of documents at a given Hamming distance. The curves quickly increase to the point where thousands of documents are equidistant. This is likely to be a property of any vector space model due to the "curse of dimensionality". Only the tails of the distribution of distances are useful for differentiation of relevant and non-relevant documents.

Approaches to near duplicate detection such as SimHash [3] use short signatures that are 64-bits in length. This only allows the few nearest neighbours to be differentiated which is adequate for near duplicate detection. This can be explained by the probability mass functions in Figures 1, 2 and 3. The x axis for 64-bit signatures will only contain 65 positions for the Hamming distances 0 to 64. As the number of equidistant documents is a function of the x value, or, Hamming distance, many documents will appear equidistant much sooner than with signatures of 1024-bits in length. The same curve has to be squeezed into 65 positions instead of 1025 positions. A duplicate is expected to be very similar to other documents it is a duplicate of, so these short signatures will suffice. In contrast, TopSig uses much longer signatures that allow for better separation for tasks such as ad hoc retrieval and clustering.

Document clustering places similar documents into groups of topically related documents. The results presented in this paper suggest that only document clusters that exist within the local neighbourhood of a vector space are interpretable. As the documents within a cluster become more dissimilar, the grouping of these documents loses its meaning for the

same reason precision at higher recall suffers in ad hoc retrieval, there are many equidistant documents that are unable to be differentiated from one another. This suggests that only a large number of smaller document clusters are meaningful. The maximum interpretable radius for a document cluster can be esimated heuristically from the distributions of estimated from the pairwise data. This heuristic is to stop at the point where the distribution starts to sharply increase. In Figure 1 this would be approximately a Hamming distance of 450, or, the point before the elbow in the left hand side of the distribution occurs.

Furthermore, TopSig is likely to be useful for increased computational efficiency of document-to-document comparisons. Examples of this include clustering, classification, filtering, relevance feedback, near duplicate detection and explicit semantic analysis. All of these tasks can exploit the left tails of the probabilty mass function distributions depicted in Figures in 1, 2 and 3. In fact, TopSig has been shown to provide a 1 to 2 magnitude increase in processing speed for document clustering [7] over traditional sparse vector representations.

The analysis presented in this paper is expected to be useful for any vector space model. It would be expected that similar behaviour would be exhibited whether comparing entire documents in the full vocabulary space of the term-by-document matrix or comparing dimensionality reduced documents in a continuous space such as those produced by latent semantic analysis, principal component analysis or random indexing.

## 7. REFERENCES


[1] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *KDD 2001*, pages 245–250. ACM, 2001.

[2] B.H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[3] M.S. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC*, pages 380–388, New York, NY, USA, 2002. ACM.

[4] C.M. De Vries, R. Nayak, S. Kutty, S. Geva, and A. Tagarelli. Overview of the INEX 2010 XML mining track. *INEX 2010, LNCS*, 2011.

[5] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[6] C. Faloutsos and S. Christodoulakis. Signature files: an access method for documents and its analytical performance evaluation. *ACM Trans. Inf. Syst.*, 2:267–288, October 1984.

[7] S. Geva and C.M. De Vries. TopSig: topology preserving document signatures. In *CIKM 2011*, pages 333–338. ACM, 2011.

[8] G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis*, 2(2):205–224, 1965.

[9] W.B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics*, 26(189-206):1–1, 1984.

[10] M. Sahlgren. An introduction to random indexing. In *TKE 2005*, 2005.

[11] A. Trotman, X. Jia, and S. Geva. Fast and effective focused retrieval. In *Focused Retrieval and Evaluation*, LNCS, pages 229–241. 2010.

[12] I.H. Witten, A. Moffat, and T.C. Bell. *Managing gigabytes: compressing and indexing documents and images*. Morgan Kaufmann, 1999.

[13] J. Zobel, A. Moffat, and K. Ramamohanarao. Inverted files versus signature files for text indexing. *ACM Trans. Database Syst.*, 23:453–490, December 1998.

# Putting the Public into Public Health Information Dissemination: Social Media and Health-related Web Pages

Robert Steele
Discipline of Health Informatics
The University of Sydney
Sydney, NSW, Australia
robert.steele@sydney.edu.au

Dan Dumbrell
Discipline of Health Informatics
The University of Sydney
Sydney, NSW, Australia
ddum7449@uni.sydney.edu.au

## ABSTRACT
Public health information dissemination represents an interesting combination of broadcasting, sharing, and retrieving relevant health information. Social media-based public health information dissemination offers some particularly interesting characteristics, as individual users or members of the public actually carry out the actions that constitute the dissemination. These actions also may inherently provide novel evaluative information from a document computing perspective, providing information in relation to both documents and indeed the social media users or health consumers themselves. This paper discusses the novel aspects of social media-based public health information dissemination, including a comparison of its characteristics with search engine-based Web document retrieval. A preliminary analysis of a sample of public health advice tweets taken from a larger sample of over 4700 tweets sent by Australian health-related organization in February 2012 is described. Various preliminary measures are analyzed from this data to initially suggest possible characteristics of public health information dissemination and document evaluation in micro-blog-based systems based on this sample.

## Categories and Subject Descriptors
D.3.3

## General Terms
Documentation.

## Keywords
Twitter, Web documents, Public Health

## 1. INTRODUCTION
The role of the Internet in enabling document retrieval and dissemination has affected vast change in the past 15-20 years. Relatively recently, searches for certain types of information were (and still are) usually done via search engines resulting in the ranked presentation of the algorithmically calculated most relevant Web documents. With the introduction and development of social media platforms however there has been some change in the discovery and retrieval aspects of Web documents.


*ADCS '12*, December 5-6, 2012, Dunedin, New Zealand.


Public health information dissemination involves communication of disease prevention and health promotion information through organized efforts. While a majority of Web-based public health information dissemination and retrieval has been done via search engines, it has been found that relevant public health documents were not always successfully located and disseminated via search engines due to the query behavior of the user [1].

The novel approach of utilizing social media for public health information dissemination and retrieval has recently been explored [2]. The rapid and widespread uptake of social networking sites (SNSs) such as Twitter allows for the sharing of public health-related information to result in more up-to-date information dissemination due to the instantaneous and 'push' nature of the application. Twitter, a widely used micro-blogging service, has characteristics that make it a useful tool for information dissemination and retrieval, such as instantaneous 'tweeting' (postings of 140-character limited updates),'re-tweeting' (forwarding of other users' tweets) and the ability to publicly interact with other users and their tweets. Unlike the case with search engines, documents in Twitter are discovered via embedded URLS in received tweets.

When dealing with public health information dissemination via Twitter, there are a number of objectives and goals that are beyond simply retrieving relevant documents that match queries based on information retrieval metrics. These co-occurring objectives include: the aim of widespread dissemination of quality health information; dissemination to targeted groups and individuals so that the information is reaching those it is relevant to; achieving 'push' communication with the ultimate goal of positively affecting the health behavior of recipients; involving users with cognate interests to interact and communicate; as well as providing up-to-the-minute information.

The purpose of this paper is to provide a preliminary overview of micro-blog-based public health information dissemination and its novel document retrieval and evaluation characteristics. A preliminary experiment to examine relationships between public health advice re-tweeting behavior and the nature and authority of the Web pages pointed to by embedded URLs in public health advice tweets is described and results presented.

## 2. BACKGROUND
URLs that are embedded in tweets represent Web links to documents that can provide the user with lengthier health-related information (usually summarized or suggested in the tweet itself). A recent study by Cui et al. found that from an analysis of one million tweets, 29.1% contained URLs [3]. However after further examination it was found that over half these URLs were 'spam'-

related. On the other hand, a study examining public health-related tweets by Australian health organizations found that a large majority of public health-related tweets included URLs and that they were also genuine (i.e. non-spam and contained appropriate information as described in the tweet) [4]. These characteristics of tweets embedding URLs for public health document dissemination and retrieval suggests the value in further research into social media-disseminated public health documents.

Throughout the literature reviewed, evaluation of public health messaging in micro-blog applications such as Twitter (and more generally social media) has not been thoroughly explored. There have been various temporal estimation methods and models introduced for information retrieval and document rankings when compared to traditional media (i.e. newspaper articles) [5]. Document and data ranking (by Web search) has recently been explored when taking into consideration the social aspects of SNS like Twitter and Facebook [6]. The authors of the study proposed a ranking system based on the characteristics and communicative relationships between SNS application users as well as the actions these users performed on Web documents.

A form of socially-based Website review was introduced called tagging, where end-users place a content and quality label on a particular document on a topic of interest. These types of technologies [7] used on various Websites can also be seen on Twitter, whereby a user can perform various actions (e.g. re-tweeting and favouriting) to provide both dissemination and opinions/evaluation of the content of the document (or tweet).

## 3. SOCIAL MEDIA-BASED PUBLIC HEALTH INFORMATION DISSEMINATION

There are a number of inter-connected and diverse characteristics of social media-based public health information dissemination. In terms of dissemination, such systems could be considered to represent a distributed health information dissemination network with the network topology and routing depending upon the 'self-organizing' activities of the human members of the social media network. This self-organizing aspect of social media is present via users first manually choosing which accounts to follow, based on their preferences for what information and accounts are of interest and relevant to them, and also in choosing from various possible actions including forwarding or re-tweeting when receiving a health-related micro-blog post.

We identify novel aspects specific to social media-based public health information dissemination to include: user role in document dissemination, public review and evaluation, known and targeted recipients, impact of population values and user-initiated content.

**User role in document dissemination:** Users are actually forwarding and hence are the parties disseminating public health information or Web documents. When users receive information that they find interesting or useful, they may re-tweet the micro-blog post including any embedded link referring to a Web document.

**Public review and document evaluation:** Actions taken in relation to the public health information received from other accounts is also a source of evaluative information. By carrying out such actions as re-tweeting or favouriting, users are also passing a form of judgment and evaluating the type, quality and/or relevance of the tweet and any document referred to. For example, favouriting a document may be making an indication about the relevance of the public health information based on the needs of the particular user, whereas re-tweeting may suggest that there is some value in that piece of public health information for followers to benefit from. These issues will be further explored in Section 5 via the preliminary experiment, analysis and results described.

**Known and targeted recipients:** Another significant characteristic of social media-based public health dissemination is that the recipients (at least the receiving accounts) of any given piece of public health information can be known. One of the challenges with mass media-based public health dissemination is that it can be hard to establish who has received a given message or how successfully targeted it is. Such capabilities in social media suggest the ability to analyze and optimize dissemination in detail across the population and also create software and analytic tools to measure and optimize public health information dissemination via social media.

**Impact of population values:** Social media-based systems also include an interplay between community interests and values, and what and how broadly information is disseminated. For example in [8] the issue of acute health risk indicated in tweets, and hence its possible relationship to feelings of social obligation, was identified as a characteristics of highly re-tweeted tweets.

**User-initiated content:** Varied types of users are also able to generate their own health information and micro-blog postings (whether this is accepted or not is based on various factors, such as user perception of the authority or interest of that piece of information). This example identifies another aspect of Twitter-based public health dissemination, whereby accepting and interacting with certain users and documents involves establishment of an informal 'network of human trust' as part of that information dissemination.

## 4. SOCIAL MEDIA vs SEARCH ENGINES

As stated, traditional forms of Internet-based access to public health information are often through the use of search engines. This could be broadly considered to be a 'pull'-based approach, where the user would discover and request information from various sources, and in this case public health Web documents. However with the introduction of SNS, the users of these services share information as the means for these documents to reach other individuals. While SNSs like Twitter support the sharing of short textual messages, they are very commonly utilized to direct others to Web pages and documents by providing a URL. Twitter is a good example of a SNS that incorporates a 'push'-based approach (see Table 1), where specified, relevant and up-to-the-minute health information is pushed to users. This may suggest the relevance of social media-based public health dissemination where health behavior modification is aimed for.

The criterion used to determine the resulting documents seen by users varies between the two systems. Search engines base their results on the user's query and identify the most relevant and authoritative pages based on those query words. In Twitter however, discovered documents are based on the type of account the user chooses to follow, and hence the quality and content of the public health information is dependent on the tweeting users being followed. This inherently includes some form of manual evaluation being done by the mass of users (see Section 5) implicit in the sharing and dissemination activities.

Social media is well suited to providing up-to-the-minute information. Up-to-the-minute information is well-suited to public health for various reasons and scenarios, such as epidemic outbreaks, health warnings, natural disasters and environmental information updates to name a few.

Another of the major differences between social media and search engine systems is the level of interaction with a public health Web document or information. Due to the open environment of Twitter, there are many mechanisms for peer feedback.

**Table 1. Comparison of Web-based public health information dissemination systems**

| | Social Media | Search Engines |
|---|---|---|
| Mode | 'Push' | 'Pull' |
| Temporal | Most recent | Various times |
| Ranking/ evaluation | Manual/human selected | Algorithmic |
| Interaction | Community and peer-based | Individual-based |
| Document description | Manual by micro-blog poster | Automated/ anchor text-related |
| Documents disseminated | Changing rapidly | Relatively constant for a given query |
| Documents received | Relevent to a topic & topic personally selected | Relevent to a query |
| Web coverage | Limited to documents shared | More comprehensive |

## 5. PRELIMINARY ANALYSIS OF A SAMPLE OF PUBLIC HEALTH RE-TWEETS

In the dissemination of public health advice via Twitter, there are a number of ways in which the broad populations of users inherently provide some form of evaluation of that information.

### 5.1 Preliminary Analysis of Sample

As part of a broader study described in [4], all tweets by health-related organizations in Australia meeting threshold criteria of at least 150 followers per account and having been sufficiently recently active, were collected and manually analyzed for the month of February 2012. There were 114 identified health-related organization accounts meeting these criteria, and these accounts produced 4787 tweets during that month.

These 4787 tweets were also categorized as being of various 'types' including for example: public health advice, organizational news, advertising, fundraising, conference and event, amongst various others. In this paper we limit our consideration to just the sub-set of these tweets that were public health advice tweets.

Across the whole 114 identified health-related organisations, there were a total of 772 (out of the overall total of 4787 tweets) public health advice tweets (re-tweeted and non re-tweeted) sent in February 2012. A total of 359 of these public health advice tweets were found to have been subsequently re-tweeted at least once. The identified Twitter accounts were also categorized into three sectors – government (16), for-profit (FP) (27) and not-for-profit (NFP) (71) organizations.

For this paper we were interested to gain preliminary insights into the relationships between public health information re-tweeting behaviour, and the Web documents pointed to by embedded URLs within the public health advice tweets, and also other preliminary characterizations of public health social media usage in terms of this sample.

Of the 359 public health advice tweets sent in February 2012, that were re-tweeted, 329 of these re-tweeted tweets contained a URL (91.6%). Notably, this is substantially higher than the proportion of re-tweeted tweets in general which contain embedded URLs (56.7%) [9].

All 359 tweets were ordered in terms of number of times they were re-tweeted. The most re-tweeted was re-tweeted 40 times, the top 25 tweets were re-tweeted seven or more times, the next 94 tweets were re-tweeted between three and six times inclusive and the remaining 240 were re-tweeted one or two times.

The Web pages indicated by the embedded URLs were manually inspected to determine the source of their content. The tweets that either did not include an embedded URL or for which this link was no longer functioning were excluded from this analysis. In the most highly re-tweeted group (seven or more times) there was some evidence of high authority for the information source of the linked-to Web pages: 40% were from leading NFPs such as the Australian Red Cross, Cancer Council NSW, the Heart Foundation etc.; 25% from government departments and 20% based on input/ content from Professors.

The second set of tweets considered were those re-tweeted three to six times inclusive. In this case only 31% of Web pages indicated by these tweets were from NFP organizations, a slightly lower percentage indicated government site pages (23%), a much lower percentage were from Professors (1.5%) and a small percentage (6%) had as their source international journal articles.

The third set, re-tweeted only once or twice showed again a decrease in NFP Web pages (26%), a decrease in government content pages (18%) and a decrease in Professors as sources of the Web page content to 1%. Interestingly there was an increased and relatively large proportion of pages in this group that used as their information source international journal articles (20%).

The above shows high authority sources such as government, well-recognized NFPs or Professors, as the sources for the content of indicated Web pages, being more prevalent for more highly re-tweeted tweets. On the other hand it also shows high authority sources such as international journal articles being more prevalent amongst lower re-tweeted tweets, counter to the possible overall trend of the embedded URLs within more highly re-tweeted tweets indicating pages with more authoritative information sources.

As an alternative analysis, to gain a simple measure of the 'authority' or 'quality' of the Web pages indicated by embedded URLs, the Google PageRank of each of the indicated Web pages (and of their domain name resulting from removal of the directory path from the end) were retrieved via the Google toolbar. Various averages of these PageRanks were then calculated. Tweets from the re-tweet set were excluded in this analysis if they did not include a URL, had a URL but it was no longer functioning or no PageRank was available for that URL.

For the most re-tweeted group (re-tweeted seven or more times) the average PageRank for the indicated Web pages was 4.25, for the set of tweets re-tweeted three or more times, the average was 3.9 and for the set of all tweets re-tweeted at least once the average was 3.38. In relation to the domain names for the top group of re-tweeted tweets the average domain PageRank was 6.52, for tweets re-tweeted three or more times, the average domain PageRank was 6.39 and for the set of all re-tweeted tweets the average domain PageRank was 6.05.

While these averages suggest that a higher PageRank was present on average for the URLs in more re-tweeted tweets, there was a low positive correlation between PageRank and re-tweet count with only $r=0.16$, with significance value of $p=0.01$.

In considering other factors with possible correlation with high re-tweeting, interestingly the number of followers of an account did not show a strong correlation with number of times re-tweeted with r= 0.243 and significance value p=0.000. Also there was a moderately strong negative correlation between total number of tweets sent from an account and PageRank of embedded URLs, r= -0.614 with p=0.000.

Other analysis of the sample data provides some preliminary insight into those doing the re-tweeting and hence carrying out the possible evaluative actions. In general, accounts of individuals were the most active re-tweeters: individual accounts re-tweeting numbers outnumbered government accounts re-tweeting for example by a ratio of ten-to-one. This suggests the large role of individuals in providing the document evaluative information. Finally, if a user refers to another on Twitter via the @ symbol, this is considered a 'mention'. There were only 51 instances of mentions in our sample of 359 re-tweeted tweets.

## 5.2 Discussion

The low r value for the correlation between PageRank and re-tweet number may be a result of various other factors impacting re-tweeting such as the actual semantic content of tweets, PageRank not being an appropriate measure of the quality or authority of public health information or possibly there being little linear relationship between page authority and re-tweet number. Previous work has also shown a not very high correlation between PageRank and the quality of health information [10].

The low r value of account follower number in relation to number of re-tweets is interesting in that it suggests that it is not just number of individuals receiving a tweet that drives numbers of re-tweets, which might be naively hypothesized, but it suggests that characteristics intrinsic to the content of the tweet may be more important in affecting the level of re-tweeting. The negative correlation between number of tweets and PageRank of embedded URLs suggests accounts sending many tweets are not tweeting URLs with high authority as indicated by PageRank.

Previous research has found that re-tweets contained a significantly larger percentage of embedded URLs when compared to regular tweets (i.e. URLs have a strong relationship with re-tweetability) [7], (56.7% and 19.0% respectively) [8]. From our sample, it may possibly be hypothesized that in relation to public health advice tweets the inclusion of URLs is more common and of even greater importance to achieve dissemination.

# 6. FUTURE RESEARCH

While public health social media networks may potentially create powerful 'self-organized' dissemination networks the effectiveness of these dissemination networks for public health needs to be further investigated.

Much further work is required to determine how effective re-tweeting and evaluative actions are in identifying and rewarding quality or relevance of public health Web pages or micro-blog posts. An immediate future step would be a more sophisticated measure of authority or quality of health information provided.

Related questions include: does such dissemination and evaluation implement a "wisdom of the crowd" behavior?; does the self-organized nature effectively route documents to individuals who are better qualified to evaluate these documents?

There are a far broader range of future research questions also. How effectively are public health tweets reaching target audiences? Given information on who are the social media recipients is available, related to this, what can be determined about a user from their connection network and postings? How well are users able to identify the relevant accounts to follow? Finally, in relation to healthcare, are users actually changing their health behavior as a result of receiving such information?

# 7. CONCLUSION

This paper has considered the novel characteristics of social media-based public health information dissemination. A preliminary study has considered the relationship between re-tweeting frequency and the authority or quality of the Web pages pointed-to by tweets and other preliminary dissemination characteristics. Future related directions of research are also identified.

# 8. REFERENCES


[1] Yang, C.C., Winston, F., Zarro, M.A. and Kassam-Adams, N. 2011. A Study of User Queries Leading to a Health Information Website:AfterTheInjury.org. In *Proceedings of iConference* (Seattle, USA, February 08 – 11, 2011). iConference'11. 267-272.

[2] Paul, M.J. and Dredze, M. 2011. You Are What You Tweet: Analyzing Twitter for Public Health. In *Proceedings of the 5th International Conference on Weblogs and Social Media* (Spain, July 17 – 21, 2011). ICWSM'11. AAAI, 265-272.

[3] Cui, A., Zhang, M., Liu, Y. and Ma, S. 2011. Are the URLs really popular in microblog messages? In *Proceedings of IEEE International Conference on Cloud Computing and Intelligence Systems* (Beijing, China, September 15 – 17, 2011). CCIS'11. IEEE, 1-5.

[4] Dumbrell, D. and Steele, R. 2013. Twitter and Health in the Australian Context: What Type of Information are Health-related Organizations Tweeting? In *Proceedings of the 46th Hawaii International Conference on System Sciences* (Hawaii, USA, January 07 – 10, 2013). HICSS'13.

[5] Efron, M. and Golovchinsky, G. 2011. Estimation methods for ranking recent information. In *Proc, of the 34th Intl. Special Interest Group on Information Retrieval* (Beijing, China, July 24 – 28, 2011). SIGIR'11, 495-504.

[6] Khodaei, A. and Shahabi, C. 2012. Social-Textual Search and Ranking. In *Proceedings of CrowdSearch Workshop* (Lyon, France, April 17, 2012). CrowdSearch'12. 3-8.

[7] Miguel, A.M., Karampiperis, P., Kukurikos, A., Karkaletsis, V., Villarroel, D. and Leis, A. 2011. Applying Semantic Web technologies to improve the retrieval, credibility and use of health-related web resources. *Health Informatics Journal*, 17, 2 (June, 2011), 95-115.

[8] Dumbrell, D. and Steele, R. 2012. What are the Characteristics of Highly Disseminated Public Health-related Tweets? In *Proceedings of the Australian Computer-Human Interaction Conference* November 26 – 30, OZCHI'12.

[9] Zarrella, D. Science of Retweets. http://www.slideshare.net/danzarrella/the-science-of-re-tweets, 2009. Date Accessed – September 15 2011.

[10] Griffiths, K., Tang, T., Hawking, D. and Christensen, H. 2005. Automated Assessment of the Quality of Depression Websites. Journal of Medical Internet Research, 7(5).

# Reordering an index to speed query processing without loss of effectiveness.

David Hawking
Funnelback Pty Ltd.
Canberra ACT 2602, Australia, and
RsCS, Australian National University
david.hawking@acm.org

Timothy Jones
Funnelback Pty Ltd.
Melbourne VIC 3066, Australia
tjones@funnelback.com

## ABSTRACT

Following Long and Suel, we empirically investigate the importance of document order in search engines which rank documents using a combination of dynamic (query-dependent) and static (query-independent) scores, and use document-at-a-time (DAAT) processing. When inverted file postings are in collection order, assigning document numbers in order of descending static score supports lossless early termination while maintaining good compression.

Since static scores may not be available until all documents have been gathered and indexed, we build a tool for reordering an existing index and show that it operates in less than 20% of the original indexing time. We note that this additional cost is easily recouped by savings at query processing time. We compare best early-termination points for several different index orders on three enterprise search collections (a whole-of-government index with two very different query sets, and a collection from a UK university). We also present results for the same orders for `ClueWeb09-CatB` . Our evaluation focuses on finding results likely to be clicked on by users of Web or website search engines — *Nav* and *Key* results in the TREC 2011 Web Track judging scheme.

The orderings tested are Original, Reverse, Random, and QIE (descending order of static score). For three enterprise search test sets we find that QIE order can achieve close-to-maximal search effectiveness with much lower computational cost than for other orderings. Additionally, reordering has negligible impact on compressed index size for indexes that contain position information. Our results for an artificial query set against the TREC ClueWeb09 Category B collection are much more equivocal and we canvass possible explanations for future investigation.

## Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval*; H.3.4 [**Information Systems**]: Information Storage and Retrieval—*Systems and Software*

## Keywords

Enterprise search; inverted files; efficiency and effectiveness; information retrieval.


*ADCS'12* December 5-6, 2012, Dunedin, New Zealand.


## 1. INTRODUCTION

Commercial search engines are subject to strong incentives to deliver fast query response, while using as little hardware as possible. Slow response degrades user experience [3] and negates the value of "instant search"[1], while inefficient retrieval algorithms increase infrastructure costs and lower profitability. If the time taken to run a query over an index of a certain size can be reduced by a factor of $f$ while keeping the hardware constant, then the number of machines needed to support a large query load is also reduced by a factor of $f$. Not only is the cost of hardware reduced but so too is the cost of data centre hosting, system administration, and electricity. Even better, for most countries, greenhouse gas emissions fall in line with reduced electricity demand. For major world wide web search engines, a factor of two in query processing efficiency could translate to billions of dollars per annum and, depending upon energy mix, to savings of the order of a million tonnes of $CO_2$ emissions.

To illustrate the importance of this point, a European Energy Agency graph[2] shows that, worldwide, approximately 500gm $CO_2$ is emitted per kWh of electricity *generated*. A typical rack-mounted server consumes a constant 250W and in the course of a year consumes $365 \times \frac{24}{4} = 2190$ kWh. Transmission losses and data centre overheads such as airconditioning push that up to a generating requirement of around 3000 kWh p.a, corresponding to about 1.5 tonnes of $CO_2$ per server. Guessing a deployment of a million servers for a hyothetical major search engine, leads to an emission estimate above a million tonnes. (In Australia, where electricity generation is heavily reliant on coal, $CO_2$ emissions in 2009 were about 80% higher at 928gm per kWh generated.[3]) These arguments also apply in smaller scale commercial search engine deployments, such as intranets in large organisations, high-volume e-commerce sites, whole-of-government search, multi-tenancy hosted website search (Sofware-as-a-Service, or SaaS), and search "in a cloud". The largest of such deployments may index of the order of a billion documents and/or process thousands of queries per second. An increase in query efficiency allowing for smaller numbers of servers clearly has a large environmental impact.

We investigate the hypothesis that smarter ordering of documents in an index can significantly improve the efficiency of query processing without reducing quality of results. We present a tool to efficiently re-order an existing index according to a permutation file and report its cost relative to indexing time. We then take two example enterprise web indexes for which we have appropriate

[1]When a search engine takes a partially submitted query, guesses a likely completion and runs it, presenting one or more result sets before the user has finished typing.
[2]`http://www.eea.europa.eu/data-and-maps/figures/co2-emissions-per-kwh-of`, accessed 26 Oct 2012.
[3]ABB *Australia: Energy efficiency report*, updated Jan 2011.

query test sets and plot query response and effectiveness against different DAAT termination points for Original, Reverse, Random, and QIE (Query Independent Evidence, i.e. descending order of static score) permutations. We conduct a similar comparison for `ClueWeb09-CatB` using an artificial query set.

# 2. BACKGROUND AND CONTEXT

## 2.1 Search engine architecture

A simple model which is adopted in essence by many search companies is to split a large collection of documents into $m$ partitions comprising roughly equal numbers of documents. This may be done by hashing on URLs. Each partition is indexed separately and the index is replicated across the $n$ query processing servers assigned to that partition. Thus we have an $m \times n$ matrix of servers. The incoming query stream is load-balanced across the $n$ rows of that matrix. Each row supports a broker function which multicasts an incoming query to all the query processors in the row and merges the results.

The optimal partition size depends upon the hardware configuration of the query processing servers, the efficiency of the ranking algorithm, the size of documents, and the design goals for response time. With 2012-era hardware, and an average response time goal of say $r_{ave} = 100$msec, a reasonable partition size might be up to around 100 million web documents. That number could be increased significantly for shorter documents such as microblog posts or database records.

The above architecture is oversimplified and takes no account of real-life complications such as query caching, real time indexing, load balancing, efficient result merging and fault tolerance. However, it is sufficiently accurate to place the present work in context.

Here we focus on efficient ranking on a single query processing server – one node in the matrix – with an index of up to 100 million web pages. If the response time goals are achieved and the server has $c$ processor cores, the whole matrix will be able to handle $\frac{c \times n}{r_{ave}}$ and each individual server will be able to handle $c/r_{ave}$ queries per second. For a 12-core server that would give a total throughput of 120 queries per second, assuming that query processing is CPU-bound and that the full benefit of parallelism can be achieved across the cores.

## 2.2 Queries and information needs

Although law firms, patent departments, investment advisers, medical reviewers and intelligence agencies have genuine needs for complex, high-recall search, here we focus on the set of queries which comprise a small number of words and no operators, and which result from an information need which can be satisfied by a small number of documents. The vast majority of queries submitted to the commercial search engines we have been discussing belong to this set.

For example, a student at a university searches for 'exam timetable' and clicks only on the link which takes them to the relevant timetable site; A computer scientist searches on the Web for 'RFC 3261' and clicks only on the Session Initiation Protocol document from IETF; A citizen searches for 'tax', and selects the home page for the national taxation office. In other words, although a typical query may find many matching documents, the searcher is most often satisfied by a very small number of them.

Queries without operators are often described as "bag of words" queries. However, modern commercial ranking functions assign higher scores to documents which match the sequence of words in the query or subsequences of it. In the following example queries from the TREC-2009 Web Track set, it seems intuitive that documents matching the query as a phrase are more likely to be useful than those containing unassociated occurrences of the query words: "mitchell college", "rick warren", "orange county convention center", "pampered chef". Accordingly, it may be better to think of "sequence of words" queries.

Since 2010, the TREC Web Track [6] has adopted a six-point relevance judging scale: {Nav, Key, HRel, Rel, Non, Junk}. *Nav* and *Key* correspond to the desired answers to homepage finding and topic distillation tasks respectively. These tasks were investigated in earlier Web Track campaigns.

Documents receiving clicks in the categories of search represented by the examples above are likely to be Nav or Key documents. Consequently, we focus our evaluations on common Web-style queries and these Nav or Key answers.

## 2.3 Ranking and query matching methods

We assume that documents are ranked by a combination of query-dependent (dynamic) and query-independent (static) scores, as is believed to be the case in most commercial search engines. Query independent scores may include link-graph measures [15, 4], access frequency [12, 13], document quality scores [16, 1], non-spam score [8], recency, and so on. The overall ranking function is typically machine-learned [23] and may combine more than 1000 features.

Sequence-of-words queries which contain more than one word $(W_1, W_2 \ldots W_n)$ may be processed either term-at-a-time (TAAT) or document-at-a-time (DAAT) [24]. In unoptimised TAAT, all documents containing $W_1$ receive a partial score, then those containing $W_2$, have their scores augmented and so on up to $W_n$. Finally, all documents with non-zero scores are sorted in descending score order. Until the last term is being processed, it cannot be known which documents satisfy the AND of the query terms. An obvious optimisation is to store postings in impact order and truncate low-impact postings, but this is potentially lossy, as truncated postings for one term may have combined with high-impact scores for another, and AND matches or even phrase matches, may be missed. Long and Suel [14] claim that the TAAT model is infeasible at large scale.

In DAAT, postings lists for terms are scanned in "parallel" and are assumed to be in document number order. It is easy to determine whether candidate documents match the AND of the query terms and whether they match the query as a phrase[4].

The DAAT model avoids the need for a large accumulator set, supports more efficient AND or Weak-AND (WAND) processing [2], allows for easier upweighting of phrases and terms in proximity, delivers higher quality results in the event of a forced timeout, and, as we shall see, supports principled early-termination. Svore et al [22] claim retrieval effectiveness gains of up to 13% through use of proximity features in Web search.

Here we restrict our discussion to the DAAT model.

## 2.4 Index formats

According to Ding and Suel [11], methods for early termination of query processing use one of three inverted file structures:

- **Document-sorted Indexes:** in which the postings in each list are sorted by document ID. This is the usual representation for DAAT-based early termination techniques.

- **Impact-sorted Indexes:** in which the documents in each postings list are sorted by their impact on ranking under e.g. Cosine or BM25. This is the usual approach for TAAT-based early termination techniques

[4] Assuming that the index includes term positions

- **Impact-layered Indexes:** in which the postings are divided in to layers according to impact, but those layers are sorted by document ID. This allows some of the benefit of impact sorting in DAAT techniques, although it comes at a compression cost due to larger gaps between document IDs.

Ding and Suel note that document-sorted indexes are less studied for early termination strategies [11]. In this paper we assume a document-sorted index, since it fits the DAAT model which is feasible at web scale.

Compression increases the size of index which can be accommodated in a given memory size, or read in a single read from disk. Much work in query processing assumes that indexes cannot be fully resident. More recently, this assumption has been revisited [9, 21], and according to [10] Google's web search indexes have been memory-resident since 2003.

In this paper we make no assumption about whether indexes are fully resident or not. The index ordering method investigated here will bring benefit in the fully resident case and also increase the locality of reference in the partially resident case. In the experiments reported here the small indexes are fully resident and the `ClueWeb09-CatB` index is not.

## 2.5 Optimising DAAT processing when static scores are used

Long and Suel [14] were the first to illustrate that early termination of DAAT processing is possible when postings are arranged in order of decreasing static score. In their work, PageRank [15] was used for the static score, and a modified cosine measure for the dynamic component. Several different pruning strategies were evaluated in terms of number of disk blocks accessed during query processing, query throughput per second, and error rate – defined as the percentage deviation from a full ranking for some $k$ documents. The strategies that best balanced a high throughput with a low error rate used two postings lists per term – the first list contains the $h$ postings that contain the highest term value according to the cosine measure, and the second list contains the remaining postings. Single-list based strategies had high throughput, but were also associated with a higher error rate.

To illustrate lossless early termination, let us assume that the final score of document $d$ is $F_d = \alpha \times D_d + (1-\alpha) \times S_d$, where all scores are normalised to $0 \ldots 1$. Thus, for example, $\alpha = 0.3$, scores are determined 70% by static factors and 30% by the extent to which the document matches the query. If, during DAAT processing, we are looking at a document $d$ whose static score is $S_d$ and the $k$-th best document found so far has a final score of $F_k$, then processing can stop if $F_k > \alpha + (1-\alpha) \times S_d$. I.e. even if a document not yet encountered achieves a dynamic score of 1.0, it will not make it into the final top-$k$.

Additional optimisation can be performed based on the dynamic score component. In the Weak-AND technique proposed by [2], an upper bound $UB_t$ for contribution to the final ranking score is recorded alongside each postings list. In the simple case, $UB_t$ is equal to the maximum score contribution from any document in the postings list for term $t$. Due to the difficulty of calculating this upper bound for phrase terms or other complex query-time combinations of postings lists, $UB_t$ can be estimated. Document scores are fully evaluated each time a match is found, and a heap of the $h$ highest scoring documents is maintained. Once the heap is full, the score of the lowest scoring document in the heap can be used in combination with $UB_t$ to faciliate early advancement or termination in postings lists where that term's score contribution cannot push the document over the score required to join the heap. A further optimisation is to record the upper bound for blocks of postings [11], which allows skipping more documents without loss of quality.

In practice, processing can terminate quite a bit earlier than the lossless point with very low deterioration of scores on evaluation measures. Our experiments explore the effect on search quality of varying the cutoff point in a basic DAAT approach, similar to the naive approach in [14]. We use this naive approach as it does not depend on the composition of the dynamic ranking function, which allows for the query time customisation and flexibility desirable in enterprise search.

In this context, lossless early termination can apply only when the static part of the ranking function used at query processing time corresponds to that used when the index order was determined. In enterprise search this is often not the case, because search profiles associated with different classes of users may assign different weightings to the static variables.

In the equation $F_d = \alpha \times D_d + (1-\alpha) \times S_d$, low values of alpha are common, but this doesn't mean that matching the query is unimportant. In the model we assume, candidate documents must generally satisfy the AND of the query terms[5]

We note that in some search engines alpha is a function of the length of the query – the longer the query the higher the value of alpha.

## 2.6 Reordering document IDs

Other studies have investigated the effect of reordering document IDs on compression. [20] investigated several different orders, and found that sorting documents by URL is cheap and results in effective compression. This is because documents with similar URLs are often topically similar, since they are located in similar directories on the same server.

Further compression is possible when context information for term frequencies or position is carried across between documents. [25] introduced a scheme using this context to achieve additional compression when indexes were sorted by URL (although they note their improvement is also likely to apply to other orderings).

## 2.7 Aims

The aims of the present study are:

- to confirm the findings of Long and Suel on enterprise scale web collections, using enterprise search test sets and a combination of static features found to be useful in enterprise search,
- to confirm the findings of Long and Suel for fully resident and partially resident indexes,
- to quantify the in-practice time costs of reordering all components of an existing index.
- to quantify the impact on compression of reordering an index by QIE instead of crawl order.

# 3. METHOD

## 3.1 Retrieval system

We use an experimental variant of a commercial retrieval system whose inverted file indexes include word position information. Postings lists are stored as variable-byte [18] encoded differences. Skip blocks are not included in the inverted files used in the experiments reported here. The index comprises many files in addition to

[5]In practice, stopwords may be removed, very long queries may be truncated and the strict AND requirement may sometimes be weakened, e.g. when there are few full-AND matches.

**Table 1: Datasets used in the experiments. Index size includes only the index files needed in the present experiments. Other files used in query suggestion, summary generation etc are not included. Note that the `ClueWeb09-CatB` dataset was indexed with options designed to limit the size of the index. For example the `gov-Whole` index includes a great deal more metadata. Average query length is in words.**

| Test | No. doc.s | index size (GB) | no. of queries | ave query length |
|---|---|---|---|---|
| `University` | 386325 | 1.2 | 134 | 1.37 |
| `gov-Popular` | 2294156 | 9.1 | 91 | 1.25 |
| `gov-Agencies` | 2294156 | 9.1 | 100 | 3.84 |
| `ClueWeb09-CatB` | 50217545 | 89.6 | 100 | 3.10 |

the basic inverted file: sorted term dictionary, files to support snippet generation, a document table recording document properties such as length, spam features, recency, URL length, inlink scores, query-independent content feature scores, web host feature scores and so on.

The results of experiments reported here are dependent on the value of $\alpha$. If it were close to one, there would be little or no value in reordering the index. If it were close to zero the value of reordering would be increased. However, arbitrary settings of $\alpha$ make no sense since they would lead to poor result quality. We used a configuration of the ranking function in which $\alpha = 0.39$ which has been shown to produce good results across eleven query test sets involving eight different enterprise document collections.

Precise details of the ranking function are not important, since it is constant across all experiments. Dynamic scores are computed in BM25-like [17] fashion, taking into account document fields and referring anchortext with additional weight for implicit-phrase and proximity features. The static score is a fixed weighting of features derived from the link graph, document URL, document recency, document quality classifiers, the host domain and the host graph.

## 3.2 DAAT cutoff

It is possible to specify a stopping condition in terms of the number of full AND matches found for the query. I.e. stop after $z \geq k$ full matches have been found. We explored how response time and result quality varied with changes in $z$ for different index orders.

In all our experiments the number of results displayed was $k = 10$.

## 3.3 Infrastructure

All experiments were run on a laptop computer with a quad-core CPU (Intel Core i7-2720QM) with a clock-speed of 2.20GHz. It was equipped with 16GB of RAM and a single 7200 RPM SATA disk drive. No advantage was taken of the four CPU cores as index reordering and query processing were run single-threaded. Larger RAM would be recommended for a production deployment but this configuration gave us the potential to explore the value of index reordering on a non-resident index.

## 3.4 Timing

Timing results reported below represent the elapsed (wall-clock) time taken to run the query test set in a single execution of the query processor. Times are elapsed times and include query preprocessing, query matching, and ranking only. Snippet generation, spelling suggestion, faceting and related query suggestion functions are not included.

## 3.5 Datasets

The datasets used are summarized in Table 1. The query / answer sets for `University` and `gov-Popular` were generated by webmasters for those organisations. Queries correspond to popular and/or business critical queries.

We had intended to use the `ClueWeb09-CatB` queries and answers from the 2011 TREC webtrack [7]. Unfortunately, once they were filtered down to exclude answers outside the Category B set, there were only ten queries with Nav or Key answers. This was considered too small to support meaningful experiments.

Instead we accessed lists of universities in the US, Canada, New Zealand, Australia, United Kingdom and Ireland. For each university whose homepage and Wikipedia entry were in the `ClueWeb09-CatB` collection, we used the name of the university as the query and recorded the homepage as a Nav answer (grade 4) and the Wikipedia page as a Key answer (grade 3). In some cases we added multiple Wikipedia URLs or multiple homepage URLs but treated them as equivalents – no extra credit for retrieving more than one of them. Many of the university homepages and some of the Wikipedia entries are not present in the category B collection and those universities were rejected. However with the addition of a handful of additional universities from China, Singapore and Denmark we reached our target of 100 queries. As shown in the table, the queries are quite long (average: 3.10 words).

Example queries (showing only one of multiple equivalent answers):

| |
|---|
| University of Otago<br>`www.otago.ac.nz/`<br>`en.wikipedia.org/wiki/University_of_Otago` |
| Stony Brook University - State University of New York<br>`www.stonybrook.edu/`<br>`en.wikipedia.org/wiki/SUNY_Stony_Brook` |
| University of Cambridge<br>`www.cam.ac.uk`<br>`en.wikipedia.org/wiki/University_of_Cambridge` |

We would be very happy to make this test set available to other researchers on request.

The `gov-Agencies` query set is another artificially constructed set in which the queries comprise the names of agencies within the government and the answers (Nav only) are the homepages of those agencies. Query lengths in this set are even longer on average (3.8 words).

# 4. EFFECTIVENESS MEASURE

For consistency with the TREC Web Track, and because the arguments in [5] are quite persuasive, we used the Expected Reciprocal Rank (ERR) measure. Our relevance grades were converted to be consistent with a grade of 4 for a Nav answer and 3 for a Key answer, and we used the same function as the TREC Web Track for maping assessor grades to gain values. Our measurement software explicitly avoids giving extra credit when multiple equivalent URLs are returned. For example, a university homepage may have multiple URLs which lead to the same content. E.g. `www.uni.edu/` and `uni.edu/`.

# 5. INDEX REORDERING

We built a tool which takes each of the many files comprising an index and makes a reordered copy of them. The new order is defined by a text file containing a permutation of the document numbers $1, \ldots, |C|$, where $|C|$ is the number of documents in the collection. The reorder command takes three arguments: old index, new index, and permutation file.

Small files which are subject to permutation are read entirely into memory. Then each record is accessed in the new order and written

sequentially to the copy. Naturally, memory space is freed after each file is permuted.

Some large files (such as the data from which snippets are generated) may be processed in windows to avoid random disk I/O. In this case, the output file is divided into windows whose size corresponds to the available physical memory. The input file is scanned sequentially once for each window. During a scan, records in the current window are read into memory and reordered. At the end of the scan, those records are written sequentially to the output file.

The order of the term dictionary does not change, but pointers from dictionary entries to the inverted file may do so. As each dictionary entry is processed, its postings list is decompressed, then permuted, then compressed and written to the output file. Finally, the pointer from the dictionary entry to the inverted file is updated.

Simple tools were used to create permutation files from the original index. The three permutations we investigated were: Reverse, Random, and QIE (Query Independent Evidence, i.e. descending order of static score).

# 6. RESULTS

We first of all report the times taken to reorder the indexes and then report the results of query processing experiments in which the DAAT cutoff value is varied.

## 6.1 Time to reorder collections

Table 2 shows that the time taken to reorder an index is a small fraction of the original indexing time. Note that times reported include the reordering of all non-temporary index files, not only the ones needed to support the stripped-down results presentation used in our query processing experiments.

## 6.2 Index size

Unlike [20, 25], we did not observe a change in index size under different document orders. Observed sizes differed by at most one or two percent. This is likely to be due to the inclusion of term position information, which does not change between orderings, and makes up the bulk of the index. Possibly using context sensitive compression for posittion information [25], or alternative index structures for matching phrases [19] would yield changes in index size after reordering.

## 6.3 Varying the DAAT cutoff parameter

Figures 1, 2, 3 and 4 show the effect of varying the DAAT cutoff value on search effectiveness (as measured by ERR on the test sets) and on computational effort. Elapsed times are the time to process the full query batch and are the median of five observations.

We used a perl script to determine for each index order, the DAAT cutoff at which a criterion level of 95% of overall maximum ERR score was achieved and to report the elapsed time corresponding to that cutoff. The results for `University` are shown in the following table, with the time ratio expressed relative to the lowest value in Column 3.

| Order | Cutoff | Elapsed time @ cutoff | Time ratio |
|---|---|---|---|
| Original | 40 | 0.132 | 1.11 |
| QIE | 10 | 0.119 | 1 |
| Random | 640 | 0.227 | 1.91 |
| Reverse | 1280 | 0.292 | 2.45 |

We see a big effect for the order of the index. Near-maximal performance is achieved at cutoff 10 for QIE order but not until 640 or 1280 for Random and Reverse orders. Indeed Reverse doesn't ever match the maximum QIE effectiveness level. Interestingly, the original index order (corresponding to approximately breadth-first crawl-order) performs relatively well, attaining the criterion at a cutoff of 40. The time ratio column shows that achieving the criterion perfermance level requires twice as much computation for the Reverse and Random orders.

For `gov-Popular` we see even larger differences between the best order and the worst:

| Order | Cutoff | Elapsed time @ cutoff | Time ratio |
|---|---|---|---|
| Original | 2560 | 0.421 | 3.53 |
| QIE | 160 | 0.141 | 1.18 |
| Random | 2560 | 0.455 | 3.82 |
| Reverse | 80 | 0.119 | 1 |

The original order for this collection was not crawl order and it seems to be particularly perverse. By reversing the order we can reduce the cost of achieving criterion performance by a factor of 3.5! In this case Reverse order slightly outperforms QIE, but the difference is only small.

The results for `gov-Agencies` shown in Figure 3 and in the table below, show a similar picture despite the fact that queries are very much longer and artificially created. Criterion performance is achieved with a much lower DAAT cutoff for QIE order than for the other orders and Random and Reverse orders require roughly twice as much computational effort to reach criterion performance.

| Order | Cutoff | Elapsed time @ cutoff | Time ratio |
|---|---|---|---|
| Original | 80 | 0.541 | 1.47 |
| QIE | 20 | 0.369 | 1 |
| Random | 160 | 0.875 | 2.37 |
| Reverse | 80 | 0.634 | 1.72 |

For `ClueWeb09-CatB` the picture is rather different to that of the other three experiments. Figure 4 shows that at very low cutoffs the QIE and Original orders achieve substantially better effectiveness than Random or Reverse. However the lines converge more quickly than for the other data sets and the performance of the QIE order relative to that of the Original falls away. Results in the following table appear to show that Original and Reverse orders can achieve criterion (95% of maximum) effectiveness with only 40% of the computational effort. The fact that the best order and its reverse achieve similar performance levels after a while seems interesting.

| Order | Cutoff | Elapsed time @ cutoff | Time ratio |
|---|---|---|---|
| Original | 2560 | 12.72 | 1.50 |
| QIE | 5120 | 20.98 | 2.47 |
| Random | 5120 | 23.56 | 2.78 |
| Reverse | 1280 | 8.49 | 1 |

In all four experiments, very close to 100% of maximum effectiveness is achieved by cutoff 5120, regardless of index order.

# 7. DISCUSSION

On the basis of the `University` experiment, it seems that there is a very clear advantage to be had from creating a favourable ordering of the index and choosing a low DAAT cutoff value. The `gov-Popular` and `gov-Agencies` experiments seem to confirm the value of this approach. However, the results for the `ClueWeb09-CatB` experiment do not show anywhere near as strong an effect for index ordering.

The QIE order does not perform as well on `ClueWeb09-CatB` as in the other experiments. A possible explanation is that the `University` , `gov-Popular` and `gov-Agencies` test sets are included among the eleven test sets used to set the weights for the static variables in the ranking function, but that `ClueWeb09-CatB` is not. Including `ClueWeb09-CatB` data among the tuning testsets

**Table 2: Elapsed time taken to reorder the indexes as a percentage of the original indexing time, including the time taken to determine the permutation. Reorder times used were the median of 3 observations.**

| Test | Original | qie | Reverse | Random | notes |
|---|---|---|---|---|---|
| `University` | 100 | 12.0 | 8.8 | 11.8 | index originally in crawl order |
| `gov-Whole` | 100 | 16.1 | 14.5 | 19.5 | 1index not in crawl order |
| `ClueWeb09-CatB` | 100 | 18.2 | 15.6 | 18.3 | index not in crawl order |

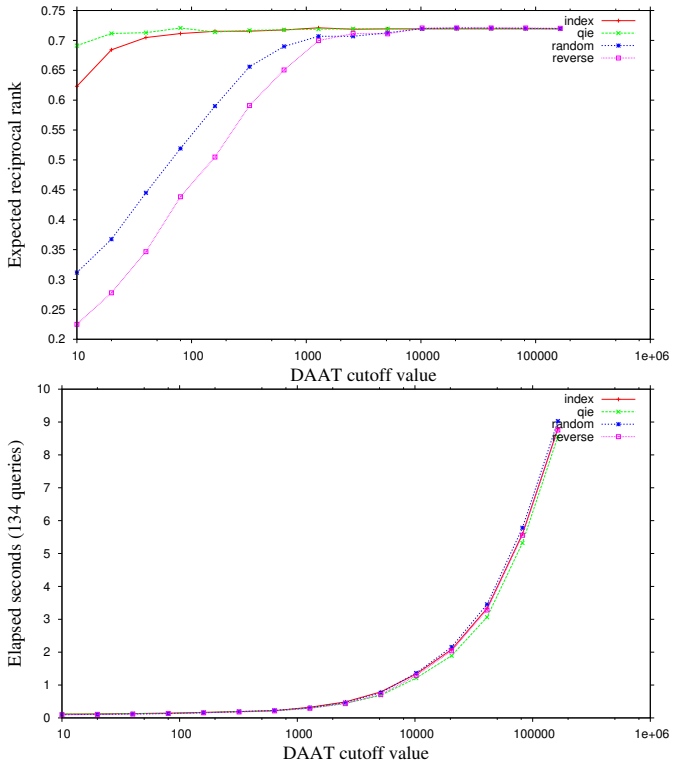



**Figure 1: Results for `University` plotted against DAAT cutoff. The upper plot shows expected reciprocal rank while the lower one shows time to process the batch of queries.**

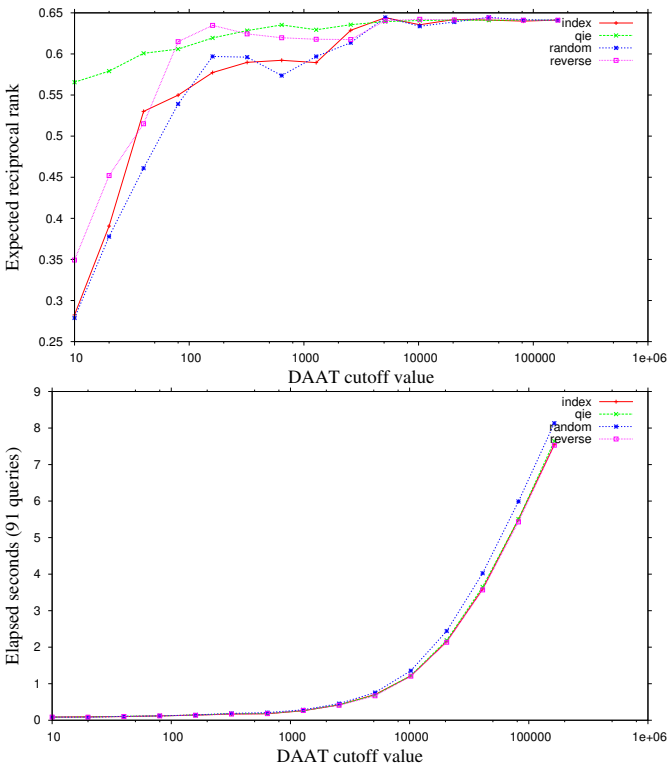



**Figure 2: Results for `gov-Popular` plotted against DAAT cutoff. The upper plot shows expected reciprocal rank while the lower one shows time to process the batch of queries.**

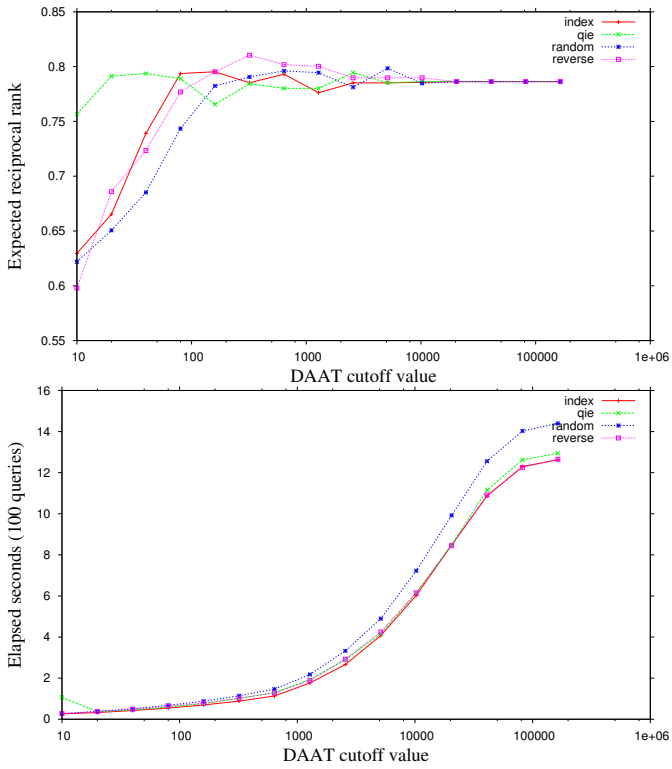



**Figure 3: Results for `gov-Agencies` plotted against DAAT cutoff. The upper plot shows expected reciprocal rank while the lower one shows time to process the batch of queries.**

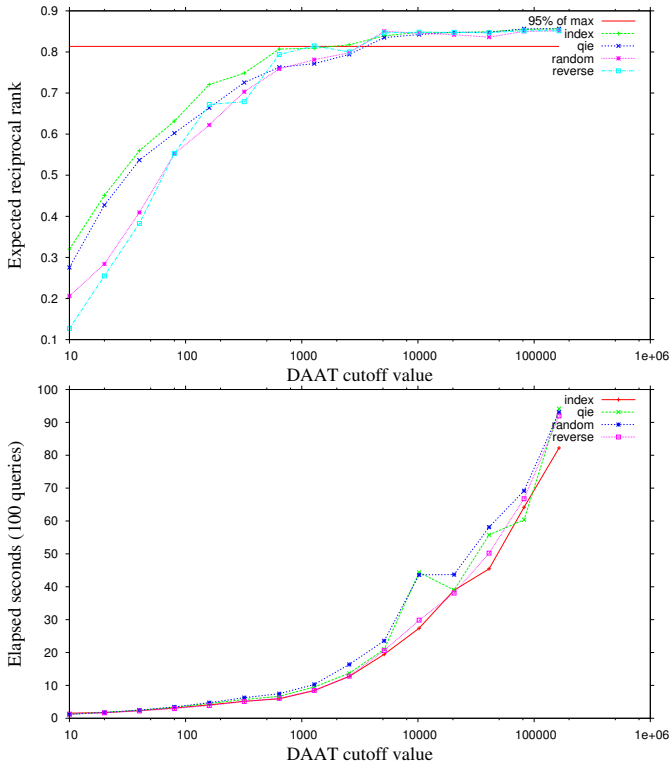



**Figure 4: Results for `ClueWeb09-CatB` plotted against DAAT cutoff. The upper plot shows expected reciprocal rank while the lower one shows time to process the batch of queries.**

may have resulted in different tunings and better performance of the QIE order in the present experiment.

Without a clear effectiveness result on the `ClueWeb09-CatB` data, we are unable to conclude anything useful about differences between resident and only partially resident indexes.

# 8. CONCLUSIONS AND FUTURE WORK

We have shown that reordering of an existing index can be achieved in a small fraction of the original indexing time, even when the size of the index is much larger than the available RAM configuration. Such reordering is of practical importance in collections where documents are gathered in an unfavourable order and where the optimal order is only determined during indexing.

Based on three of our experiments it appears that reordering an unfavourably ordered index allows near-full effectiveness to be achieved with only a fraction of the computational effort needed to fully index the collection. These findings essentially confirm the results of Long and Suel, in a range of different conditions, using a combination of features found to be effective across a range of enterprise search collections.

Further work is clearly needed to understand the different outcome of our fourth experiment. Useful follow up experiments include using a different query set or learning a better QIE order for the `ClueWeb09-CatB` data.

Crucially, we found that reordering indexes that include term position information does not affect the compressed index size. This allows for arbitrary orderings without affecting memory residency.

# 9. REFERENCES


[1] M. Bendersky, W. B. Croft, and Y. Diao. Quality-biased ranking of web documents. In *Proceedings of WSDM 2011*, pages 95–104, New York, NY, USA, 2011. ACM.

[2] A. Z. Broder, D. Carmel, M. Herscovici, A. Soffer, and J. Zien. Efficient query evaluation using a two-level retrieval process. In *Proceedings of CIKM 2003*, pages 426–434, New York, NY, USA, 2003. ACM.

[3] J. Brutlag. Speed matters for Google web search. Technical report, Google, June 2009. `services.google.com/fh/files/blogs/google_delayexp.pdf`.

[4] P. Calado, B. Ribeiro-Neto, N. Ziviani, E. Moura, and I. Silva. Local versus global link information in the web. *ACM TOIS*, 21:42–63, January 2003.

[5] O. Chapelle, D. Metzler, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of CIKM 2009*. ACM, 2009.

[6] C. Clarke, N. Craswell, and E. Voorhees. TREC 2012 web track guidelines, 2012. `http://plg.uwaterloo.ca/~trecweb/2012.html`.

[7] C. L. Clarke, N. Craswell, I. Soboroff, and E. Voorhees. Overview of the TREC 2011 Web Track. In *Proceedings of TREC 2011*. NIST, 2011.

[8] G. V. Cormack, M. D. Smucker, and C. L. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Inf. Retr.*, 14(5):441–465, Oct. 2011.

[9] J. S. Culpepper, M. Petri, and F. Scholer. Efficient in-memory top-k document retrieval. In *Proceedings of SIGIR 2012*, pages 225–234, New York, NY, USA, 2012. ACM.

[10] J. Dean. Challenges in building large-scale information retrieval systems: invited talk. In *Proceedings of WSDM 2009*, pages 1–1, New York, NY, USA, 2009. ACM.

[11] S. Ding and T. Suel. Faster top-k document retrieval using block-max indexes. In *Proceedings of SIGIR 2011*, pages 993–1002, New York, NY, USA, 2011. ACM.

[12] S. Garcia and A. Turpin. Efficient query evaluation through access-reordering. In *Proceedings of the Third Asia conference on Information Retrieval Technology, AIRS 2006*, pages 106–118, Berlin, Heidelberg, 2006. Springer-Verlag.

[13] Y. Liu, B. Gao, T.-Y. Liu, Y. Zhang, Z. Ma, S. He, and H. Li. BrowseRank: letting web users vote for page importance. In *Proceedings of SIGIR 2008*, pages 451–458, 2008.

[14] X. Long and T. Suel. Optimized query execution in large search engines with global page ordering. In *Proceedings of VLDB 2003*, pages 129–140, 2003.

[15] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford, January 1998. `dbpubs.stanford.edu:8090/pub/1999-66`.

[16] M. Richardson, A. Prakash, and E. Brill. Beyond PageRank: machine learning for static ranking. In *Proceedings of WWW 2006*, 2006.

[17] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proceedings of TREC-3*, pages 109–126, November 1994. NIST special publication 500-225.

[18] F. Scholer, H. E. Williams, J. Yiannis, and J. Zobel. Compression of inverted indexes for fast query evaluation. In *Proceedings of SIGIR 2002*, pages 222–229. ACM Press, 2002.

[19] D. Shan, W. X. Zhao, J. He, R. Yan, H. Yan, and X. Li. Efficient phrase querying with flat position index. In *Proceedings of CIKM 2011*, pages 2001–2004, New York, NY, USA, 2011. ACM.

[20] F. Silvestri. Sorting out the document identifier assignment problem. In *Proceedings of ECIR 2007*, pages 101–112, Berlin, Heidelberg, 2007. Springer-Verlag.

[21] T. Strohman and W. B. Croft. Efficient document retrieval in main memory. In *Proceedings of SIGIR 2007*, pages 175–182, New York, NY, USA, 2007. ACM.

[22] K. M. Svore, P. H. Kanani, and N. Khan. How good is a span of terms?: exploiting proximity to improve web retrieval. In *Proceedings of SIGIR 2010*, pages 154–161, New York, NY, USA, 2010. ACM.

[23] K. M. Svore, M. N. Volkovs, and C. J. Burges. Learning to rank with multiple objective functions. In *Proceedings of WWW 2011*, pages 367–376, New York, NY, USA, 2011. ACM.

[24] H. Turtle and J. Flood. Query evaluation: strategies and optimizations. *Inf. Process. Manage.*, 31(6):831–850, Nov. 1995.

[25] H. Yan, S. Ding, and T. Suel. Inverted index compression and query processing with optimized document ordering. In *Proceedings of WWW 2009*, pages 401–410, New York, NY, USA, 2009. ACM.

# Comparing scanning behaviour in web search on small and large screens

Jaewon Kim[†] Paul Thomas[§] Ramesh Sankaranarayana[†] Tom Gedeon[†]

[†]Research School of Computer Science
The Australian National University
{jaewon.kim, ramesh.sankaranarayana, tom.gedeon}@anu.edu.au

[§]CSIRO ICT Centre
Canberra, Australia
Paul.thomas@csiro.au

## ABSTRACT
Although web search on mobile devices is common, little is known about how users read search result lists on a small screen. We used eye tracking to compare users' scanning behaviour of web search engine result pages on a small screen (hand-held devices) and a large screen (desktops or laptops). The objective was to determine whether search result pages should be designed differently for mobile devices. To compare scanning behaviour, we considered only the fixation time and scanning strategy using our new method called 'Trackback'. The results showed that on a small screen, users spend relatively more time to conduct a search than they do on a large screen, despite tending to look less far ahead beyond the link that they eventually select. They also show a stronger tendency to seek information within the top three results on a small screen than on a large screen. The reason for this tendency may be difficulties in reading and the relative location of page folds. The results clearly indicated that scanning behaviour during web search on a small screen is different from that on a large screen. Thus, research efforts should be invested in improving the presentation of search engine result pages on small screens, taking scanning behaviour into account. This will help provide a better search experience in terms of search time, accuracy of finding correct links, and user satisfaction.

## Keywords
Scanning behaviour, small screen, Trackback

## 1. INTRODUCTION
Accessing the web on mobile devices is becoming increasingly popular.[1] In this context, the following question is important: on small devices, what is users' scanning behaviour when they search for information on the web? In this study, scanning behaviour is defined as the users' actions on each element of the search engine result pages, such as seeking strategies, fixation, scanpath, click, or scroll. This question is of interest because understanding scanning behaviour is invaluable for improved interface design or for obtaining more targeted metrics for evaluating the retrieval performance [3, 5, 8, 12]. If there is any difference in users' scanning behaviour on small and large screens, then we should consider designing the presentation of results differently for each sized devices.

One method for understanding scanning behaviour is analysing transaction log files that have information about click-through, queries, and scrolling interactions between users and search engines [4, 9, 13, 17]. Another approach uses diary studies to investigate the use of web search engine with individual interviews [18]. Beyond these earlier studies, eye tracking seems to facilitate our understanding of users' attention, because their gaze can show when they are paying attention to elements of web search engines, moment by moment [3, 5, 7, 8].

Much research has been conducted on users' scanning behaviour by using eye tracking to determine where and how people look at web search results. Several studies have classified users' scanning behaviour according to gaze patterns. Klöckner et al. [11] found that 52–65% of participants used what they call a 'depth-first' strategy (the subjects scanned only the links above the selected link), 11–15% used a 'breadth-first' strategy (the subjects looked through all the links before making a decision and selecting a link), while the remaining 20–37% showed a 'mixed' strategy (looking ahead a few results past the selected link). Aula et al. [1] defined two kinds of evaluation patterns. They suggested that 54% of subjects who scanned less than half of the visible results were 'economic' evaluators, and that the others had an 'exhaustive' evaluation style. Dumais et al. [7] extended the classification and defined three clusters—Economic-Results, Economic-Ads, and Exhaustive—to identify users' scanning patterns when viewing the results of major commercial search engines that include additional links such as sponsored links or advertisements. According to their results, the Economic-Results and the Economic-Ads groups tended to spend more time on the first three results than did the Exhaustive users (68%, 61%, and 53%). In


*ADCS '12*, December 5–6, 2012, Dunedin, New Zealand.


[1] A report from 2010 indicates that even though desktop PCs or laptops are used, more than 40% of Australians mainly use mobile phones to access the web.

Source: "Australian mobile internet usage doubles", http://thenextweb.com/au/2010/09/23/australian-mobileinternet-usage-doubles, (Retrieved May. 06, 2012).

addition, the total fixation time of each group showed that the Exhaustive participants reviewed the results most slowly.

Numerous researchers have investigated scanning of behaviour features to study how users normally scan the elements of search results. Several research studies have examined eye fixations related to scrolling, and scanning behaviour above and below the selection [8, 10]. Their results suggest that subjects rarely scan below the selected link except when the link is at the page fold (when users often scan further), and that the scanning direction is from top to bottom. Buscher et al. [3], who used behavioural log data in a commercial web search engine instead of eye tracking, found that users who spend shorter time on search result pages tend to inspect just a few results, scroll less, and use fast mouse movement. Lorigo et al. [12] described two types of tasks, informational and navigational, that may impact task completion time. They found that users tend to spend more search time on informational than on navigational tasks.

Given that we have to display search results differently on small screens, if users' scanning behaviour is different from that on large-sized screens, there may be other implications for the display of search results. Only a few studies of eye tracking on small screens have been performed. Drewes et al. [6] investigated gaze interaction for controlling applications on a handheld device using the dwell-time method and gaze gestures. Further, Nagamatsu et al. [14] investigated a remote gaze tracker for small devices with stereo gaze tracking. Recently, text interaction and reading performed on an actual mobile touch screen device was analysed by Biedert et al. [2]. However, no study has compared users' scanning behaviour during web search on small and large screens. Therefore, it is not clear how users read typical displays of search results on mobile devices, and whether these displays can be improved.

In this study, we conducted an eye-tracking study with 32 participants who completed 20 tasks on large and small screens and we focused on the relation between scanning behaviour and the screen size by resizing the web browser. We measured fixation time to investigate users' search performance and attention, adopted one classification method from previous work, and defined a method we call 'Trackback' to see how far subjects look ahead prior to making the first selection.

## 2. EXPERIMENT

With an eye-tracking instrument that provides users' scanning behaviour of web search result pages, we recorded gaze data for each of the 32 participants who completed search tasks on a large as well as a small display screen.

### 2.1 Tasks

Each participant completed 20 search tasks. Inspired by Lorigo et al. [12], we used 10 tasks of each of the two kinds of tasks (informational and navigational) to investigate any influence of the task type. Each task showed a task description and a predefined query obtained from Dumais et al. [7] modified to optimize local participants' understanding (see Table 1), and search results were retrieved from the Google mobile search engine, from which we removed images, maps, or related links so that all tasks showed the same kinds of elements. All results pages were cached as local files in the system and relevant pages were shown when subjects chose the links. The tasks were very simple, needing only 1–2 min to complete. According to the search results from the Google search engine, 18 of the predefined result pages contained a relevant solution within the top 3 results, with the other two including a relevant result in ranks 4–6.

**Table 1: Examples of task descriptions and queries. Nav denotes navigational task and Info denotes informational task.**

| Task Description | Initial Task Query | Task Type |
|---|---|---|
| Find the official homepage of the Canberra casino and hotel in Canberra. | Canberra Casino | Nav |
| Go to the homepage of the Canberra Cavalry baseball team. | Canberra cavalry baseball | Nav |
| What is the standard length of a cue used for playing billiards? | billiard cue size | Info |
| How many spikes are in the crown of the Statue of Liberty? | statue of liberty crown spikes | Info |

### 2.2 Design

The participants were divided into four groups of eight, and the tasks were arranged in two sets: set 1 consisted of informational tasks 1 to 5 and navigational tasks 1 to 5, and set 2 consisted of the remaining tasks (i.e., set 1 consisted of I1N1I2N2I3N3I4N4I5 N5 and set 2 of I6N6I7N7I8N8I9N9I10N10, where 'I' denotes an informational and 'N' a navigational task). Each subject performed both task sets, one on a large screen and one on a small screen, and both the set order and screen order were counterbalanced across subjects. In other words, subject 1 performed task set 1 (TS1) on the large screen and then task set 2 (TS2) on the small screen, followed by subject 2 performing TS2 on the large screen and then TS1 on the small screen, and so on (see Table 2). Therefore, each task was distributed 32 times (16 times on each screen size) over the participants. In other words, the eight subjects in each group performed the tasks in exactly the same order, and faced the two screen sizes in the same sequence. Finally, we assigned 10 areas of interest (AOIs) on each search result page to investigate users' gaze and fixation. Each AOI corresponded to a search result, i.e., a clickable link along with its snippet text and a URL.

**Table 2: Examples of design for each group. L denotes a large screen and S denotes a small screen.**

| | Task set, order and screen size |
|---|---|
| **Group 1** | TS1 on L, and then TS2 on S |
| **Group 2** | TS2 on L, and then TS1 on S |
| **Group 3** | TS1 on S, and then TS2 on L |
| **Group 4** | TS2 on S, and then TS1 on L |

## 2.3 Procedure

First, all participants listened to an introduction to the experiment, and then practiced two sample tasks on each size screen until they were familiar with the system. Their head was then fixed on a chinrest to ensure higher eye gaze detection accuracy and the eye tracker was calibrated using 5-point calibration. Next, we preceded the first task description with an initial query and then showed the result page; this procedure was repeated until task number 20 according to an automated schedule. A time notice of 3 min was given after starting each task, after which the subjects were free to either spend more time to find the answer or move on to the next task. Typing a query was not allowed to prevent the subjects from looking at the keyboard. However, they could continue to the next page of results or follow links from the list of results. The participants were allowed to ask for task description if they did not understand the task sufficiently. At the end of the experiment, the subjects were asked to complete a questionnaire about their web search experience (the questionnaire results are still being analysed, and are not discussed in this paper). The experimental run time was approximately 30 min for each participant.

## 2.4 Participants

35 subjects (19 male) between 18 and 50 years, from various disciplines and recruited on campus at a local university, participated in the eye-tracking study. All subjects were experienced in web searching and were very familiar with the Google search engine. We excluded the results from three participants for technical reasons (e.g., stability problems with eye tracking).

## 2.5 Experimental setup

All search results were obtained from the Google mobile search engine and displayed in Internet Explorer 8. Creating new tabs or new windows was prohibited. Eye gaze was recorded by Facelab 5 with a desk mounted 17" LCD monitor and with a chinrest, and data analyses were performed using Eyeworks software.[2]

The large screen ran at 1280 × 1024 pixels as the default resolution of the monitor. Because of users' movements and the obstruction of the camera by the hand used for touching a screen, it is difficult to record the point at which a user is looking [14]. To simulate the small screen of a mobile device, we used the same monitor but with a browser limited to a 320 × 480 pixel window. To compare the effects of different screen sizes, we adopted the same font size on both types of browsers. With these settings, in the case of the large screen there was no fold (see Figure 1) and the ten search results were clearly visible without needing any

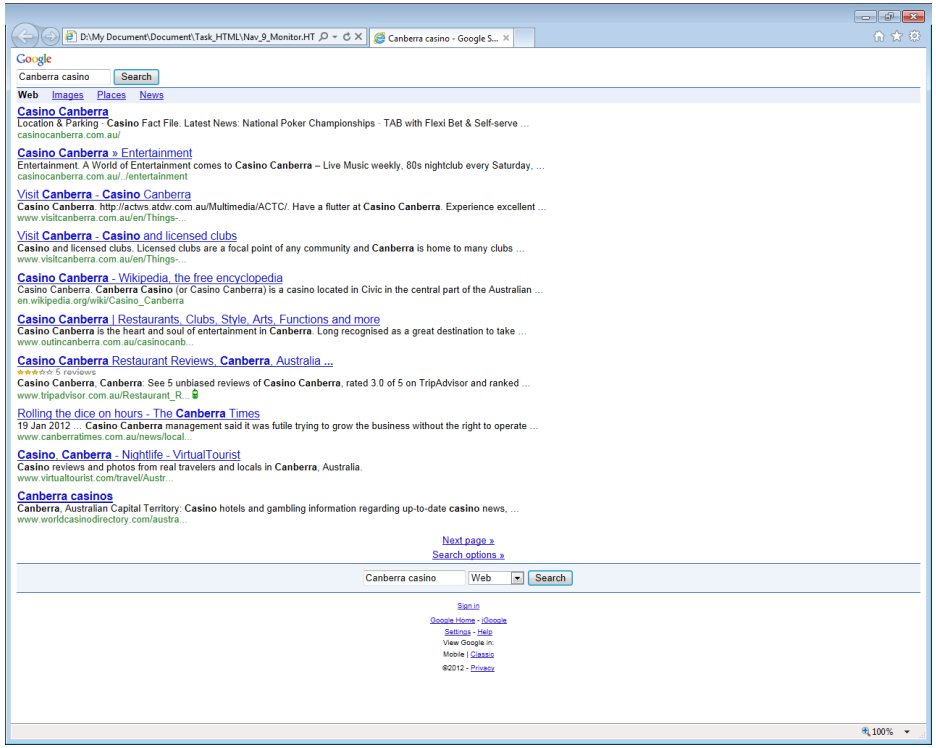



**Figure 1. Task on the large screen.**

user control of the browser, whereas the page fold was normally between positions 3 and 4 of the search results on the small screen (see Figure 2).

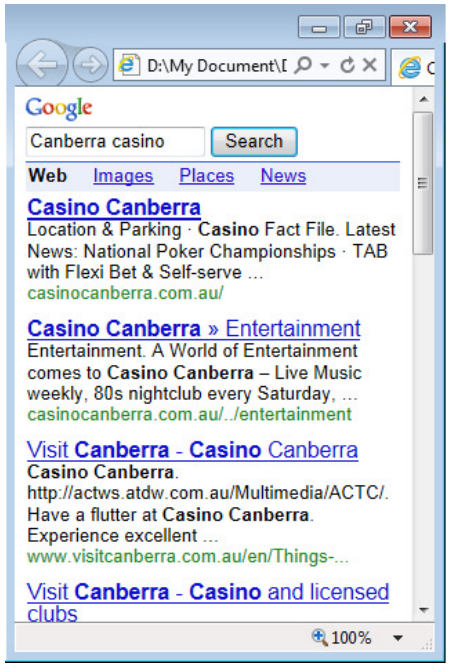



**Figure 2. Task on the small screen.**

## 3. RESULTS

Our data consist of gaze data from 640 queries (320 queries on the large screen and the same on the small screen, 160 queries of each informational and navigational task on each sized screen.) We adopted all of the 32 users' data for the results. In this paper, we focus only on fixation time, scanning strategies up to the first click, and Trackback, which we defined to describe how much users scan beyond the selected links. The other data, such as task completion time, saccades, scanpaths, and the questionnaire, are still being analysed. To study users' eye gaze in detail with the relatively small font sizes of the web search result pages on a mobile search engine, the fixations were recorded if a gaze lasted at least 75 ms and if the gaze locations were close to each other (within a radius of five pixels) using the built-in algorithms of Eyeworks software.

[2] The website: http://www.seeingmachines.com/product/facelab/

### 3.1 Fixation time

Eye fixations are useful for comparing users' scanning behaviours, because they indicate the point at which the user is looking and the fixation time represents the user's interest in each AOI or the difficulty of tasks [15, 16]. A comparison among the total fixation durations on each AOI may provide information about the usability of the interface as well as the users' efforts and search performance. The results showed slight differences between screen sizes and task types: the total fixation time up to the first click on the small screen was about 15% longer than on the large screen: 1337 s versus 1166 s, although this is not statistically significant with ANOVA $F = 3.36$, $p = 0.07$. The average fixation time for the informational tasks was slightly longer than for the navigational tasks on both sizes of screen: 3.91 s and 3.38 s on the large screen, and 4.56 s and 3.8 s on the small screen. These differences are also not statistically significant. This difference was particularly pronounced considering the top ranks: the total time spent on AOIs 1 to 3 for navigational tasks on the small screen was about 20% (100 s) shorter than it was for the informational tasks. The results according to screen sizes and task effects imply that users can easily find the link they need on a large screen in navigational searches.

Figure 3 shows the results when normalizing for total gaze time. After the first link, the AOI-normalized percentages on both screen sizes decrease sharply. However, the proportion of time spent on the first AOI on the small screen was much higher than that on the large screen (48% versus 39%) and fixations on the periphery, e.g., a query box, category tabs, or blanks between AOIs, on the large screen showed about 7% more subject attention. Even if we do not consider the proportion on the periphery, the proportion of AOIs 1 and 2 was about 4% higher on the small screen, whereas the proportions of all the other AOIs were higher on the large screen. Participants tended to spend more time on results ranked 1 to 3 when using the small screen than when using the large screen (76% on the small screen versus 67% on the large screen). This result indicates that the links ranked more than three on the small screen received very little users' attention, even though they spent a longer time overall on the small-sized screen.

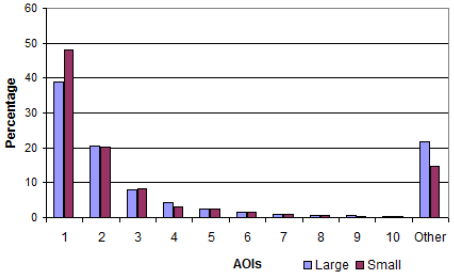



**Figure 3. AOI-normalized time spent viewing each AOI [%]**

### 3.2 Scanning strategies

We examined scanning strategies for the initial pages of search results. Even though the classification of Aula et al. [1], i.e., economic and exhaustive evaluators, seemed to be well defined

and adequate for providing more invaluable comparisons, we decided to adopt the strategies of Klöckner et al. [11], because Aula et al.'s strategy did not seem suitable for the few visible links on a small screen. If there are a few search result links, such as three or four, an economic evaluator would be defined as someone who scans only one or two of the search results.

The depth- and breadth-first strategies of Klöckner et al. [11] are useful abstractions of users' decision patterns: 'depth-first' users follow a promising link immediately; 'breadth-first' users read all their options exhaustively before clicking; and 'mixed' users read ahead but to a smaller extent. We adopted this distinction when analysing our data. Table 3 shows the total count and proportion of participants' scanning behaviours differentiated by the three kinds of strategies recognised by this approach. The table shows that subjects tended to use the depth-first strategy on the large screen slightly less than on the small screen (116 versus 131, i.e., 36% versus 41%). Instead, on the large screen they used the breadth-first strategy twice as much as on the small screen. The distributions of strategies are significantly different across screen sizes ($\chi^2 = 11.89$, df = 2, $p < 0.01$), but the count and proportion of the mixed strategy is almost the same on both screen sizes.

**Table 3. Choice of scanning strategy on both screen sizes**

| | Large | | | Small | | |
|---|---|---|---|---|---|---|
| | Depth | Mixed | Breadth | Depth | Mixed | Breadth |
| Total | 116 | 184 | 20 | 131 | 179 | 10 |
| % | 36 | 58 | 6 | 41 | 56 | 3 |

This result seems to imply that there is not a big difference in users' strategy between screen sizes. However, the majority of cases are 'mixed', with some degree of reading ahead, and therefore, this classification is perhaps hiding some differences.

## 3.3 Trackback

To examine the behaviour as the 'mixed' strategy in greater detail, we define 'Trackback' as the difference in ranks between the selected link and the farthest link observed. For example, if a subject looked as far as AOI 7 and then clicked AOI 3, the Trackback value is 4. We were able to consider only the farthest link because all our users scanned from top to bottom as in the previous study [8, 10]. With Trackback, we can scrutinize differences within the mixed strategy; the higher the Trackback, the greater is the extent to which links are observed. This method has some similarity to the scanpath analysis method used in a previous study [7, 12, 15]. However, Trackback is unique in that it summarizes the amount of additional effort users make before selecting links.

Across all users, overall there is a very significant difference; the Trackback value on the large screen is about 54% higher than on the small screen (mean 1.95 ranks/user/task versus 1.27; $t = 3.78$, df = 601, $p < 0.001$).

To examine the change in the Trackback value from large to the small screens, we calculated each participant's difference in the Trackback value between the large and small screens: the difference in Trackback for each user equals the sum of Trackback values on the large screen minus the total Trackback value on the small screen. Figure 4 illustrates the difference for each participant. Points above the x-axis represent a higher Trackback value (more looking ahead) on the large screen and points below the x-axis represent higher Trackback on the small screen. 21 users have high Trackback on the large screen whereas only 11 have high Trackback on the small screen. Therefore, screen size certainly has an effect on Trackback (ANOVA $F = 13.01$, $p < 0.001$), thereby affecting scanning behaviour. Thus, the Trackback value on a large screen is normally higher than that on a small screen.

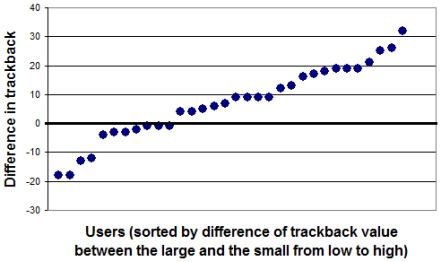



**Figure 4. Distribution of difference in Trackback between both sizes of screen. Points above the x-axis represent higher Trackback on the large screen.**

# 4. DISCUSSION AND FUTURE WORK

We have presented a study to investigate how users' scanning behaviour is different between a small and a large screen. This study showed differences in users' time spent and, by using the Trackback method, their scanning strategies on each screen size. First, from the AOI fixations in our data, we can see that users tended to take slightly longer time to decide the first selections on the small screen, although the difference is not significant. If this effect holds up under further investigation, it is probably because the small screen interface is less comfortable than the large one. In addition, the need to scroll to see more results affects the fixation time. Conversely, the difference between task types is clear: the informational tasks take more effort and time, as found in a previous study [12]. Moreover, the participants showed higher proportions of fixation time on the top three AOIs on the small screen than on the large screen, but there was no such effect associated with task type. This means that the proportions on each AOI are not influenced by task types, but by screen size. This can also be explained by the fact that the page fold is located around results 3 to 4 on the small screen, and users hardly use the scroll bar: 76% of fixations on the small screen were within the top three results, and only 9% beyond these.

Second, in the comparison between the small and large screens in terms of depth- and breadth-first strategies, users implemented more breadth-first strategy and less depth-first strategy on the large screen than on the small screen. This seems to be because of the scrolling required on the small screen, with fewer lists of results showing on the initial screen, i.e., ten results versus three or four results on the large screen versus the small screen, respectively. The classification is not entirely useful since so many participants use a "mixed" strategy and this hides real differences.

Lastly, the results of the Trackback method for observing the difference in users' choice of mixed-strategy between the large and small screen show that the average Trackback on the small screen is 1.27 per task, whereas it is 1.95 on the large screen. In addition, only about 13% of subjects showed a large negative Trackback, where the sum of Trackback values is less than –10. This indicates that subjects tend to look over more items on the large screen. We believe this means that on a large screen, user gather more information before selecting a result: they may be being more careful and checking their selection before committing to it.

There is definitely a difference in users' scanning behaviour on differently sized screens. Therefore, we should contemplate improving the presentation of web search engine result pages on small devices separately to provide users' with better search experience, even though several kinds of users' scanning behaviour have been studied on large-sized screens. The results may suggest that web interface designers or developers need to investigate the optimum presentation of search result pages for the small screen to facilitate less scanning as well as fast search time.

In further studies, first, we plan to analyse in detail the data of this experiment to reveal the difference in scanning behaviour on both screen sizes. Subsequently, since this experiment demonstrated that users tend to spend more time in spite of less scanning on a small screen, our next step will suggest an improved presentation design of web search engines on small devices by studying the relations between visible contents such as snippets, URL, or font sizes and users' scanning behaviour.

## 5. REFERENCES


[1] Aula, A., Majaranta, P. and Raiha, K-J. 2005. Eye-tracking reveals personal styles for search result evaluation. In *proceeding of the 2005 IFIP TC13 international conference on Human-Computer Interaction* (INTERACT'05), Maria Francesca Costabile and Fabio Paternò (Eds.). Springer-Verlag, Berlin, Heidelberg, 1058-1061.

[2] Biedert, R., Dengel, A., Buscher, B. and Vartan, A. 2012. Reading and estimating gaze on smart phones. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (ETRA '12), Stephen N. Spencer (Ed.). ACM, New York, NY, USA, 385-388.

[3] Buscher, G., Dumais, S. and Cutrell, E. 2010. The good, the bad, and the random: an eye-tracking study of ad quality in web search. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '10). ACM, New York, NY, USA, 42-49.

[4] Cohen, W., Shapire, R., and Singer, Y. 1999. Learning to order things. In *Journal of Artificial Intelligence Research,* 10, 243–270.

[5] Cutrell, E. and Guan, Z. 2007. What are you looking for?: an eye-tracking study of information usage in web search. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (CHI '07). ACM, New York, NY, USA, 407-416.

[6] Drewes, H., De Luca, A., and Schmidt, A. 2007. Eye-gaze interaction for mobile phones. In *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology* (Mobility '07). ACM, New York, NY, USA, 364-371.

[7] Dumais, S., Buscher, G., and Cutrell, E. 2010. Individual differences in gaze patterns for web search. In *Proceedings of the third symposium on Information interaction in context* (IIiX '10). ACM, New York, NY, USA, 185-194.

[8] Granka, L. A., Joachims, T., and Gay, G. 2004. Eye-tracking analysis of user behavior in WWW search. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '04). ACM, New York, NY, USA, 478-479.

[9] Jansen, B. J. and Spink, A. 2006. How are we searching the World Wide Web? A comparison of nine search engine transaction logs. *Information Processing and Management*, 42(1), 248–263.

[10] Joachims, T., Granka, L., Pan, B., Hembrooke, H. and Gay, G. 2005. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '05). ACM, New York, NY, USA, 154-161.

[11] Klöckner, K., Wirschum, N. and Jameson, A. 2004. Depth- and breadth-first processing of search result lists. In *CHI '04 extended abstracts on Human factors in computing systems* (CHI EA '04). ACM, New York, NY, USA, 1539-1539.

[12] Lorigo, L., Pan, B., Hembrooke, H., Joachims, T., Granka, L. and Gay, G. 2006. The influence of task and gender on search evaluation and behavior using Google. *Information Processing and Management*, 42(4), 1123-1131.

[13] Mat-Hassan, M. and Levene, M. 2005. Associating search and navigation behavior through log analysis. In *Journal of the American Society for Information Science and Technology*, 56(9), 913-934.

[14] Nagamatsu, T., Yamamoto, M., and SATO, H. 2010. MobiGaze: development of a gaze interface for handheld mobile devices. In *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems* (CHI EA '10). ACM, New York, NY, USA, 3349-3354.

[15] Pan, B., Hembrooke, H.A., Gay, G.K., Granka, L.A., Feusner, M.K., Newman, J.K. 2004. The determinants of web page viewing behavior: an eye-tracking study. In*Proceedings of the 2004 symposium on Eye tracking research & applications* (ETRA '04). ACM, New York, NY, USA, 147-154.

[16] Poole, A., & Ball, L. J. 2005. Eye tracking in human-computer interaction and usability research: Current status and future. In *Prospects, Chapter in C. Ghaoui (Ed.): Encyclopedia of Human-Computer Interaction. Pennsylvania: Idea Group, Inc.*

[17] Silverstein, C., Henzinger, M., Marais, H. and Moricz, M. 1998. Analysis of a very large AltaVista query log. SRC Technical note #1998-14. On-line at http://gatekeeper.dec.com/pub/DEC/SRC/technicalnotes/abstracts/src-tn-1998-014.html.

[18] Teevan, J., Alvarado, C., Ackerman, M.S., and Karger, D.R. 2004. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (CHI '04). ACM, New York, NY, USA, 415-422.

# Explaining difficulty navigating a website using page view data

Paul Thomas
CSIRO
paul.thomas@csiro.au

## ABSTRACT

A user's behaviour on a web site can tell us something about that user's experience. In particular, we believe there are simple signals—including circling back to previous pages, and swapping out to a search engine—that indicate difficulty navigating a site.

Simple page view patterns from web server logs correlate with these signals and may explain them. Extracting these patterns can help web authors understand where, and why, their sites are confusing or hard to navigate.

We illustrate these ideas with data from almost a million sessions on a government website. In this case a small number of page view patterns are present in almost a third of difficult sessions, suggesting possible improvements to website language or design. We also introduce a tool for web authors, which makes this analysis available in the context of the site itself.

**Categories and Subject Descriptors:** H.5.4 [**Information Interfaces and Presentation**]: Hypertext and Hypermedia
**General Terms:** Human Factors; Measurement
**Keywords:** Web documents

## 1. INTRODUCTION

For a large number of organisations, clear communication on the web is important: there is an imperative to help visitors find the information they want, lest they go elsewhere (e.g. to a competitor) or use some other, more expensive, channel (e.g. telephoning a helpdesk). Clearly, it can be a great help to understand site performance, visitor characteristics, and visitors' movements and experiences online.

As usual, if we want to understand this we have several options available. Involving users directly by observing them in action, administering surveys and questionnaires, or running interviews has the advantage of rich results accompanied by explanations of users' thought processes and responses; however, these techniques demand significant time and money and it is hard to get quick updates following website changes. The common alternative is to use records already available to understand a site: this is the goal of web analytics.

Typical web analytics is based on a web server's transaction logs. These are sparse, and contain only trace data, but they are readily available. From this, analytics packages[1] can provide an abundance of detail about a site, such as the number of visitors and where they are from; the number of page impressions; types of browsers; or the number of people who perform some action, such as making a purchase or clicking an advertisement (see e.g. Jansen [4] for an overview). These measures represent the mechanics of the web, but do not directly speak to user satisfaction, confusion, or other experiences.

In this work we attempt to bridge the two worlds, with analysis that is cheap, fast, and based on real usage; but that also provides some insight past a tally of pages visited.

We believe that server logs—aggregated appropriately—can provide insight into users' experiences online as well as the more mechanical view. We are interested in understanding what it is that this recorded behaviour can tell us about user experience, and how it can help web authors identify and fix problem spots in their sites. By contrast to conventional web analytics, we are trying to (1) *find common behaviour* that may (2) *explain why users are struggling* to find information; then (3) *present that* information to web authors in a way they find useful.

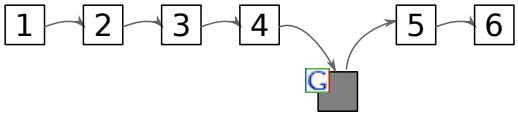
## 2. OUR APPROACH

We approach this problem through analysing user sessions as recorded in web server logs. Logs provide very simplistic data—they record only page loads, not interaction with the page, and cannot make important distinctions such as that between the time a user spends reading and the time they spend away from the screen. Despite this, they are easy to obtain for any website, and they can be collected without any extra tools on users' (or publishers') computers.

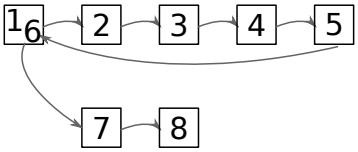
Rather than individual page views, we consider complete sequences of views—"sessions". This is for two reasons: first, there is very little information in a single recorded page view. Second, sessions describe the whole of a user's experience on a site—or as close as we can get via standard logs—which for us is more important overall than their interactions with a single page.

[1]Popular packages include those from Google (`www.google.com/analytics`), Yahoo! (`web.analytics.yahoo.com`), Clicky (`getclicky.com`), and WebTrends (`webtrends.com`). Open-source offerings include Piwik (`piwik.org`).


*ADCS '12,* December 5-6, 2012, Dunedin, New Zealand.




(a) A "swapping" session is one that includes one or more page views on our site (page views 1–4 in this example), a visit to a search engine (between 4 and 5), then one or more views on our site (5 and 6). The first portion (1–4) we call the "pre-swap" phase.




(b) A "circling" session is one that includes a sequence of page views (here 1 to 5), then a return to a previous page (6). The first portion (1–5) we call the "pre-circle" phase.

Figure 1: Two indicators that a user is having difficulty navigating a website: "swapping" and "circling" sessions.

For example, a user may visit some page $p$ but only after issuing a search and visiting several other pages. That search, and the other pages visited, suggest something about what the user was trying to do with $p$ and what task they had in mind; if we only look at transactions immediately preceding $p$, rather than the whole session, we would miss these clues. Similarly, the final page in that session may be the one that satisfied the user's information need. Knowing where that was, and not just which page immediately followed $p$, might provide clues for site design.

Our approach has three parts. First, we label those sessions where a user is having trouble: to do this, we need to identify some signals of navigation difficulty that we can extract from web server logs (Section 3). Second, we examine the logs to find patterns of behaviour that are relatively common in these sessions, ideally ones that happen as early as possible: that is, we need to find patterns that help to explain why some users have trouble (Section 4). Finally, we expose this to web authors to help them refine their sites (Section 5).

Section 6 discusses other approaches and related work.

## 3. LABELLING SESSIONS

We base our session labelling on web logs from a large Australian government agency. These logs include use of both the agency's intranet and external sites, and individual users can be distinguished [12].

In around 5% of cases, agency staff started on one of the two sites, spent some time navigating around, then switched to the other. From discussions with staff, discussions with web authors, and from examining these sessions we believe that in a large number of these cases staff were struggling to find information on the first site—probably because they were looking in the wrong place.

The first parts of these sessions must have involved some level of frustration. Comparing behaviours in these parts with behaviours in simpler sessions did show up some differences. Two behaviours in particular, "swapping" and "circling", were much more common when a user was about to change sites. The first is defined as follows:

> A *swapping* session is one in which a user browses our website, exits to a search engine, performs a search, then returns to our site (Figure 1a).

We believe this is clear evidence of navigational trouble: rather than follow links on the site, the user has found it easier to use an entirely separate tool.

Note that if the first page view in a session is the only one referred from a search engine, that session is not "swapping" by definition. Also, since we use the HTTP `Referrer:` header to recognise swaps, we are unable to detect cases where a user swaps to a search engine but never returns, for example because their information need was satisfied by the engine's snippet. Nor are we able to recognise swaps where the HTTP header is absent (about 13% of cases in the data we have examined). Our count of swapping sessions is therefore in some regard an under-estimate.

We call the part of a session before the swap the "pre-swap" phase.

Our second signal is "circling":

> A *circling* session (Figure 1b) is one in which a user views a number of pages, then retraces their steps and re-views an earlier page.

This may indicate, for example, that a particular link did not lead to the expected information; or that information the user needs is split between several pages with no obvious path between them.

Any number of repeated visits, to any page (not just the first), marks a circling session. The part of a session before the first re-view we call the "pre-circle" phase.

There are some circumstances where circling might be expected—for example, users may return again and again to a list of open jobs—so this is a weaker signal than swapping and must be interpreted with some knowledge of the site and its users. The signal is still useful, however, as the illustrations in Section 4 show.

The two behaviours are therefore suggestive, if not perfectly diagnostic, of difficulty navigating a website. In the present work we use these as indicators of "struggling" sessions, which are sessions where a user is having trouble finding the information they need. We label as struggling those sessions that are swapping, circling, or both.

Further signals, based on the data in server logs, are certainly possible (for example, one might consider reloading a page to mean something). One can also easily imagine signals from augmented logs—perhaps injecting JavaScript to log mouse movements or scrolling, for example [5]. In this work we are concentrating on logs available anywhere, from any server; crucially, this means our analysis does not depend on any modification to the site or to the server itself.

## 4. FINDING PATTERNS

If we believe that swapping is a sign of a user's navigational difficulty, it would be useful to know whether there are certain user behaviours, or certain pages, that tend to provoke it—and similarly for circling. For example, people who look at

certain pages may be more likely to swap out to Google to find what they are looking for, rather than follow the authors' cues; or people who follow certain links may be more likely to retrace their steps to gather information or to try another path. In particular, we ask:

> Given logs from a web server, are there patterns of user behaviour in the pre-swap or pre-circle phases that predict swapping or circling sessions?

That is, are there behaviours that tend to occur before circling or swapping, and are relatively more likely in these sessions? For example, in Figure 1b, is there some feature of user behaviour in 1–5 (or 1–4, or 2–5, etc.) that tells us the user is likely to circle back and try again?[2]

If the answer is "yes", then we believe these patterns could be a good clue to what is confusing a user. Of course we do not attempt to replace expert analysis with an automated tool: that said, being able to extract accurate patterns should help web designers and authors identify troublesome parts of their website, particular pages, or even particular links, and edit or redesign them. In this section we present simple techniques to do this, and briefly describe the sorts of patterns we look for at present.

The techniques are general and we believe they can work on a range of websites and can provide useful pointers for authors and analysts. We illustrate them, however, with a case study using data from almost a million sessions on a large government website. Based on the presence of swapping or circling, sessions were partitioned into "good" and "struggling" sets; the objective then is to find differences between the two that suggest where and why users have trouble.

## 4.1 Illustrative data

The website is that of a large government agency (not that of Section 3) that administers a range of programmes and has dealings with around a third of the population. The site is aimed mostly at individual citizens and residents, but it includes sections for other audiences such as businesses and professionals; it also contains a large number of forms and booklets in PDF format. This site serves several audiences and is intended to replace face-to-face or telephone enquiries as much as possible, so simple navigation is important. Authors have put some effort into arranging pages so there is a single page to answer most expected needs, and so that page can be found quickly.

We obtained one week's worth of log files from the site, representing just under one million sessions. URLs were normalised, and records that appeared to be robots were removed (including records of sessions more than two hours long, or requesting more than 1000 pages).

Table 1 summarises the cleaned data. In this table, "pages in site" counts pages actually visited, after minimal normalisation; the server hosted more pages, but these were not visited during the week. A "user" is a combination of browser and IP address, and a "session" is any sequence of page views from the same user, with no more than 30 minutes between consecutive views. "Struggling" sessions, those that swapped or circled, were truncated to the pre-swap or pre-circle phase, which means in these cases the records are of users' behaviour as they are just starting to have difficulty. (From conversations with the relevant web authors and other staff, we are confident that the two signals discussed above are appropriate for this site as well.)

[2]Note that this is an off-line problem, and we are not trying to predict or detect a user's difficulty and intervene in realtime. Even if we could manage the extremely high precision needed to intervene only when it's actually needed, it's not clear what form that intervention should take. In this work we are only interested in analysis for web developers.

| | |
|---|---|
| Log length | 7 days |
| Pages in site | 7,345 |
| Users | 748,099 |
| Sessions | 990,600 |
| Good sessions | 82% |
| Struggling sessions | 18% |
| Mean views, good sessions | 1.4 ± 1.3 |
| Mean views, pre-swap/circle | 2.2 ± 2.2 |
| Mean duration, good sessions | 0:54 ± 3:32 |
| Mean duration, pre-swap/circle | 2:08 ± 5:15 |

Table 1: Summary statistics. Views and durations are shown as mean ± one standard deviation.

## 4.2 Session characteristics

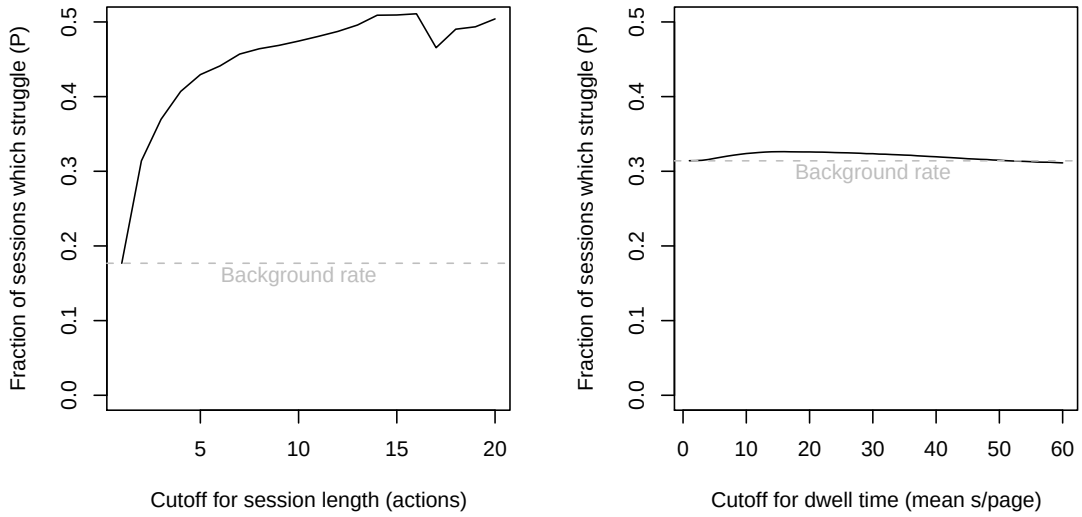
One trend is immediately apparent from Table 1. Struggling sessions are longer, in page views and time (despite being cut off just before the swap or circle), and the difference is significant (one-tailed Welch's $t$ test, $p \ll 0.01$). They are also much more variable than good sessions. An obvious implication is that it could be useful to examine sessions with many page views, sessions that take a long time, or both.

*Illustration.* We illustrate the idea with our government website data. Figure 2a plots the fraction of sessions that go on to circle or swap (precision, vertical axis) against minimum session length (horizontal axis). There is clearly an effect: users who are viewing their third page in a session have a 37% chance of going on to swap or circle, more than double the base rate, and this rises to almost 50% at 13 page views before levelling off.

Unfortunately, this is of limited use. Most sessions on this site are short, so although a long session is more likely to lead to the behaviours we associate with navigation difficulty we are not able to explain many struggling sessions this way. (With a cutoff of five pages, recall is only 12%: that is, only one in eight struggling sessions will be captured.)

Elapsed time was not useful to distinguish good from struggling sessions, again on this data. However, since struggling users tend to view more pages (Table 1), we also considered classifying sessions based on the time spent on each page (Figure 2b). (Note that since at least two page views are needed to derive dwell time, this is restricted to sessions with two or more page views. The background rate is therefore about 31%, not 18%.)

As expected, sessions with lower mean dwell times are more likely to belong to users who are having trouble and are about to swap out to an external search engine or to retrace their steps. When dwell times approach one minute per page, struggling sessions are less likely. This is consistent with past observations on both search and browsing behaviour [3, 6, 10]. Gains over the baseline are not large, however: the




(a) Struggling sessions as a function of session length, in page views: the fraction of sessions which go on to swap or circle if they have $x$ or more page views

(b) Struggling sessions as a function of dwell time: the fraction of sessions which go on to swap or circle if the mean dwell time is $\geq x$ s/page

Figure 2: The rate of occurrence of struggling sessions, as a function of session properties.

correlation between the chance of struggling and mean dwell time is poor.

## 4.3 Patterns of page views

A simple rule based on session length does detect struggling sessions early, but—at least in our example—it is not useful for authors or designers. It does not capture many struggling sessions (that is, recall is low). More importantly, it has little explanatory value: even when we know a user has had trouble, it is not clear where in the session they may have become confused. Instead, we consider slightly more complex patterns, which have the advantage of providing more explanation. In particular, we start by examining sequences of page views.

For each session in each of the "good" and "struggling" sets, we enumerate all sequences of page views—so for the three-page session $\langle x, y, z\rangle$ we would consider sequences $\langle x\rangle$, $\langle y\rangle$, $\langle z\rangle$, $\langle x, y\rangle$, $\langle y, z\rangle$, and $\langle x, y, z\rangle$. (In the examples here we ignored sequences of length five or more, and as always in the "struggling" set we only consider pages before the first swap or circle.)

Counting where these sequences appear allows us to calculate, for each sequence, the number of struggling sessions featuring the sequence; the number of good sessions; the precision P of this sequence (that is, the proportion of sessions with this sequence that do show signs of struggling); and the recall R of this sequence (the proportion of all struggling sessions that have this pattern). We can also calculate $p$ for the hypothesis that the sequence was no more common in struggling than in good sessions ($\chi^2$ test with Yates's continuity correction).

A sequence that is significantly more common in struggling than in good sessions, with reasonable precision and recall, may help explain why and where visitors have difficulty. We illustrate this below.

*Illustration.* There are 1573 sequences in our sample data where P $> 0.5$ and $p < 0.05$, some of which are shown in the top part of Table 2.

Some patterns are very precise: that is, a very high proportion of sessions exhibiting some patterns go on to signal difficulty. This is a strong clue that the pattern somehow explains users' confusion. Unsurprisingly, the most precise signals tend to be very particular. For example, in 77% of cases users with the pattern of row 1 (Table 2) went on to struggle, more than four times the expected rate; but the pattern was only seen in 22 sessions. Understanding this pattern and fixing the website accordingly would most likely be very helpful (high P), but would only fix a small number of sessions (low R).

In some cases a single page view correlated well with our signals of difficulty. The second example in Table 2, "claim form for card $x$"[3], is a case in point: 64% of users who looked at that page went on to circle or swap. This should suggest to an author that the form and surrounding information needs work. Quite likely the form itself is not self-contained, and further information is needed to fill it out (so users have to backtrack); or perhaps links to the form somehow attract users who don't in fact want it.[4] Note that conventional web analytics, based purely on counting page views or with very simple models of sessions, could not distinguish this troublesome page but could only report 1025 successful downloads.

Row 3 ("programme $y$ overview $\rightarrow$ programme $y$ eligibility") is typical of many of this type, with precision from 0.52 to 0.61

[3]These examples are anonymised.

[4]We could distinguish these two cases, and further explain users' difficulty, by looking at the page views immediately prior. If there is some page $p$ such that the transition $p \rightarrow$ "claim form for card $x$" is relatively common in struggling sessions, then the link on page $p$ is worth investigating. The software described in Section 5 supports this sort of ad-hoc analysis.

| | Page view pattern | $n_s$ | $n_g$ | P | R |
|---|---|---|---|---|---|
| 1. | income and asset tests → income tests → payments | 17 | 5 | 0.77 | $< 0.01$ |
| 2. | claim form for card $x$ | 652 | 373 | 0.64 | $< 0.01$ |
| 3. | programme $y$ overview → programme $y$ eligibility | 210 | 137 | 0.61 | $< 0.01$ |
| 4. | forms starting with "f" | 185 | 177 | 0.51 | $< 0.01$ |
| | Subsite view pattern | $n_s$ | $n_g$ | P | R |
| 5. | paper publications | 5103 | 9172 | 0.36 | 0.04 |
| 6. | audience $z$ → paper publications | 926 | 1283 | 0.42 | 0.01 |
| 7. | payments → factors affecting rates and eligibility | 2461 | 3470 | 0.41 | 0.02 |
| 8. | payments → payments (different pages) | 20147 | 30487 | 0.40 | 0.15 |

Table 2: Sample patterns which tend to occur in a struggling session. $n_s$ and $n_g$ are the number of occurences in the pre-struggle ($s$) or good ($g$) set; P is precision, i.e. the fraction of sessions with this pattern which go on to circle or swap; R is recall, the fraction of struggling sessions which contain this pattern.

for different programmes. This suggests a general problem. This pattern may appear if, for example, there's no link to apply for a programme from the eligibility page—having found they're eligible, people would have to backtrack to find out what to do next. Casual inspection of several "eligibility" pages suggests that is in fact the case, and suggests a simple fix for web authors to improve users' experience.

Finally, row 4 suggests that people have trouble navigating long lists. In fact there are 22 such lists of forms, for every letter except "g", "o", "x", and "z"; sessions passing through any one of these pages will struggle with probability well above normal. Again, this is a strong hint that some redesign is needed, although in this case the "better" design is not quite so immediately clear.

Other patterns were also apparent in our test data—for example, many common sequences involved different pages about asset testing or financial arrangements, information that would be clearer if presented on a single page. Several other patterns involved users switching between pages on two very different programmes that happen to have similar target audiences: this suggests users may be confusing the two and getting lost.

Note that the patterns we extract do correlate with struggling sessions to an interesting degree, and in many cases do suggest changes to the website. In general however the patterns could be *causes* or merely *symptoms* of confusion: for example, users viewing an FAQ page are likely already lost, and "fixing" the FAQ may not help. It may be possible to tell these two cases apart, if one pattern consistently occurs before or after others. At any rate, an analyst must still apply their knowledge of their website and users.

## 4.4 Patterns of subsites

The patterns of page views we extract with the method above can suggest particular navigation difficulties, but the same analysis can be run on different sequences. In particular, in many cases a web site will comprise several subsites, and if there are interesting patterns in subsite visits this might suggest problems with entire themes or topics as well as individual pages or links. With some definition of "subsite", which will vary for each web publishing or serving technology, the techniques above can be employed unchanged.

*Illustration.* Again we illustrate the idea with data from our government site. In this case, web authors took some care with URLs, and it was possible to define "subsite" simply as a page's containing directory (so a page `a/b/c.html` would be in the subsite `a/b`). Subsites defined this way were each specific to a topic, a particular segment of the audience, or a particular task.

In this data, there are indeed particular subsites that tend to lead to circling, or to people swapping out to a search engine. The bottom half of Table 2 contains examples, where $P$ is significantly higher than the background rate ($p < 0.05$).

Again these patterns suggest areas where users are having difficulty. For example, 36% of people who visit the "publications" subsite (PDF forms of paper publications) went on to swap to a search engine or doubled back to find something else (line 5), rising to 42% if they came from a page specific to audience $z$ (line 6). (Recall that the base rate is 18%.) Either they are unable to find the single correct form, or several forms are needed. In either case redesign might be called for. Visitors going from "payment information" to "factors affecting rates and eligibility" struggle 41% of the time (line 7). It seems likely that the directories of paper forms are confusing; and, further, that they are more confusing for audience $z$ than for other audiences.

Unlike page-based patterns, some of these subsite-based patterns have high recall. Movements between any two pages in the "payments" subsite (line 8) occur in 15% of all struggling sessions before the point of swapping or circling, with 40% precision. Patterns such as this might suggest an entire section of the website is laid out in a way that doesn't correspond with users' expectations—for example, maybe information that belongs together is split between several pages, or the language used doesn't match users' expectations. Closer examination of pages in this subsite would prove fruitful.

## 4.5 Combinations and triage

Patterns of page views or subsites can be fairly precise—that is, they can correlate well with signals of struggling sessions and provide some indication why users are having difficulty—but they tend to capture few sessions. In our illustration the single best page view pattern has recall of only around 8% and most have recall much less than 1%. Individual subsite patterns capture more sessions, but in general recall is still low.

It makes sense for web authors and analysts to focus their attention on those pages, subsites, or patterns that explain as many struggling sessions as possible: if analysis

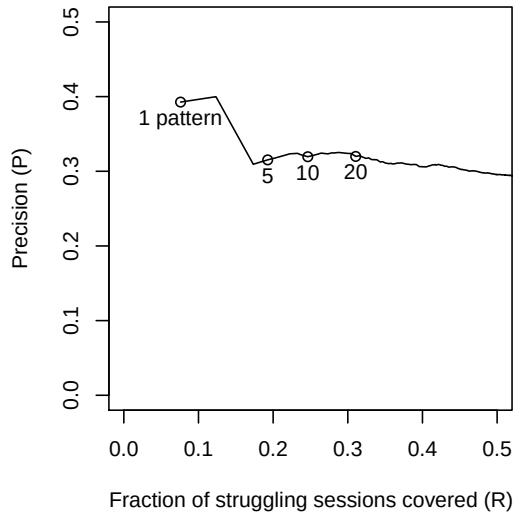



Figure 3: Precision and recall as more page view patterns are considered. Five patterns suffice to capture 19% of struggling sessions, and ten capture 25%, still with good precision.

suggests a fix, this will maximise the number of visitors who benefit. We do this by presenting *combinations* of patterns, to increase recall, and ordering them by *impact*, to help with this triage. In particular, we suggest a simple rule. First, take all those patterns that occur significantly more frequently before swapping or circling; second, prune the list to keep only those with reasonable precision; finally, order them from highest to lowest recall. Web authors should pay attention to the patterns at the head of this list.

*Illustration.* In our example, we set the threshold for "reasonable" precision to 0.2—that is, we discard patterns with precision $< 0.2$. The remaining patterns, ranked by recall, give the precision/recall curve in Figure 3.

Recall at the head of the list is good. Considering as few as five page view patterns covers almost a fifth of struggling sessions: clearly authors will not be able to fix all the difficulties in all these sessions, but any improvements will go a long way. With ten patterns, recall is 0.25; and with twenty patterns, recall is 0.31 with precision still 0.32. It seems reasonable to ask web authors, who know their site, to look at five to twenty patterns, and any improvements based on these patterns could improve many users' experiences.

# 5. EXPOSING INFORMATION FOR WEB AUTHORS

The ideas above—session-level web log analysis, finding signals of "struggling" sessions, and highlighting visitor patterns that are statistically more likely in such sessions—are instantiated in prototype software, presently deployed with the agency of Section 4.1. Again, the goal is not to automate anything: we still need people who understand their site and its users. Rather, we hope to present our analysis in a way that is useful for web authors charged with developing and maintaining a site.

As well as a backend that digests web server logs, the "latte" software includes several tools for web staff. The most important of these is the "web authors' sidebar", illustrated in Figure 4. The sidebar shows highlights of the analysis and pops up whenever a page on the relevant site is being viewed: the key idea is to integrate the analysis directly into the website, so analysts and authors can see the relevant numbers in context, while seeing what their users see. This contrasts with other tools, where the analysis is on a separate web page, spreadsheet, etc., and extra effort is needed to switch back and forth and make connections with what users see. Web authors have been using the sidebar as they move quickly about their website, looking closer when the data seems interesting.

The sidebar includes summary statistics of sessions through the page, and the proportion of those that are struggling according to the signals of Section 3. (Latte also uses two other signals, "long" and "slow" sessions, not described here.) If the fraction of struggling sessions is significantly higher than expected, either overall or by any one of those signals[5], it is marked with a "caution" sign, "**!**" (labelled (a) in Figure 3). This should suggest to the analyst that the page itself, or a pattern including this page, is causing visitors difficulty.

The sidebar also highlights other data that we think are worthy of attention. For instance, where we can glean search terms from HTTP `Referer` headers they give an explicit indication of what users were thinking of in a session. Any mismatch between the terms here and the terms on a page is likely to indicate difficulty, for example caused by visitors using different terminology or conflating two different ideas. Again, we mark with "**!**" those search terms which, when used in a session that included the present page, tend to occur when users are struggling (labelled "(b)").

As each page is viewed we also present the most common prior and following pages—that is, the most common patterns of length 2 that include the present page. Patterns that are significantly more likely to be in a struggling session are again marked (see label (c)). Showing the data in the context of the page itself makes these patterns more explicable: for example, instead of showing URLs or page titles we can show the text of appropriate links. In our experience this combination of flagging short patterns that correlate with navigational difficulty, and presenting them in context, often provides strong hints on awkward wording or confusing link structure.

We also include a tally of session endpoints (not shown in Figure 4), again for sessions passing through the current page, and mark those that are suspicious. The intuition is that the final page is often where users gather the information they need: presenting this to analysts can help clarify users' intentions in the same way as search terms.

A more detailed report is also available for each page, including session entry points, more patterns, and other data.

Finally, latte presents a list of "worst offenders". The "worst offenders" list provides a triage function similar to that described in Section 4.5: it ranks individual pages according to the number of struggling sessions through each page, provided the overall proportion is significantly higher than the background rate. Again, this helps web analysts focus on the fixes that will help the most people.

The prototype is in ongoing development, and we are working on ways to expose data on longer sequences, subsite-level

[5]To stop the signs being overwhelming, we have found it useful to raise the threshold for our hypothesis tests and only flag those cases where $p < 0.005$.

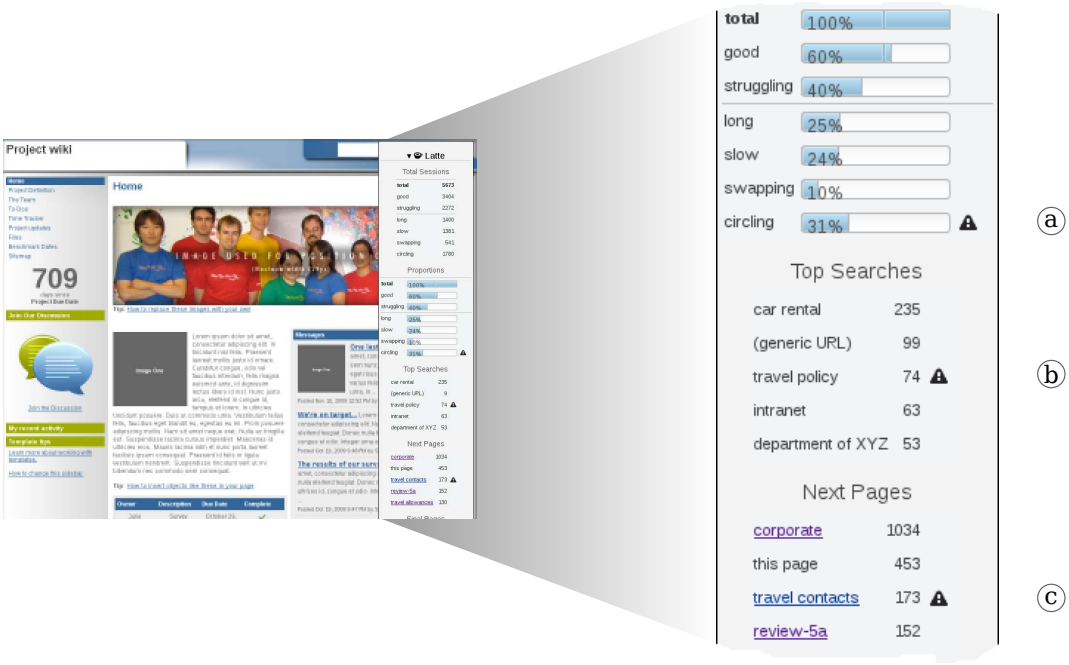



Figure 4: The "web authors' sidebar" in latte provides hints while an analyst is viewing a webpage. Here, (a) sessions that include this page are significantly more likely to circle than sessions on the whole; (b) users who search for "travel policy" in sessions through this page are significantly more likely to struggle; and (c) the sequence ⟨ this page → travel contacts ⟩ is significantly more likely in a struggling session. (Page names and search terms are anonymised.)

analysis, and trends in over time. Nevertheless, on feedback to date the tool is useful, and staff at our collaborating agency have used it to prompt changes to both navigation and wording of their site.

## 6. RELATED WORK

Other research has suggested similar uses for web log analysis, and we briefly sample some related work here.

The approach we suggest is somewhat similar to Fox et al.'s "gene analysis" [3], where abstract codings of sequences were used to help predict success at the end of a session. In this work, by contrast, we are trying to explain difficulty and we report very particular sequences—down to an individual page or even an individual link.

Sequence mining algorithms learn common patterns from recorded sequences, such as server logs (a recent survey is that by Mabroukeh and Ezeife [7]). Spiliopoulou et al. [11] use such techniques to extract abstract sequences of page views by a website's "customers" and "non-customers", and compare the two sets for leads on how to increase conversions. This is a similar idea to ours, although we are interested in a different partition, but the focus is different: while Spiliopoulou et al. involve web analysts closely in pre-processing and pattern mining, we suggest a lighter-weight, more generic tool that is appropriate for information gathering as well as transactional tasks.

Other approaches include that of Nakayama et al. [8], who suggest that pages with similar text should be visited in the same session and that a gap between textual similarity and co-visitation suggests a need for re-structuring. This relies on the intuition that visitors should read a number of pages on the same topic, but it is not clear that this is true in the web sites we have considered.

There is a rich literature on search log analysis (see e.g. Silvestri [9] for a recent survey). Our logs represent browse, not search, behaviour, so typical measures such as query reformulations or ranks of clicked results are not directly applicable; however, in future work we hope to align browse and search logs to extract further patterns and explanations.

## 7. DISCUSSION AND CONCLUSIONS

The trace data collected in web server logs is extremely limited, but does contain signals that we believe are indicative of a user's difficulty navigating a website. Using this to partition sessions into "good" and "struggling" sets lets us find patterns that occur relatively frequently in the latter, and which may help explain users' difficulty.

The algorithms described here are simple, but (as illustrated) are effective in capturing a large proportion of troublesome sessions and providing actionable hints for web analysts. While the particular patterns extracted here will not generalise across sites, the abstract techniques make very few assumptions about site layout, content, or user behaviour and should generalise well. This is comparable to other work on implicit feedback [2], which is generalisable and is easy to use without deep knowledge of a particular site; however, our technique provides more advice on where trouble spots are and what users are thinking.

(Although they cannot generalise across sites, the patterns here do generalise across *time* to some extent. Using data from another week later in the same month, the top five patterns of Section 4.5 still cover 33% of struggling sessions with 15% precision.)

We are building some of this analysis into a tool for web authors. The "web authors' sidebar" analyses server logs and shows the results in the context of a web page, while that page is loaded in a browser (Figure 4). Page transitions or search terms that are significantly more likely in struggling sessions are highlighted with "caution" signs—in this example, there are two transitions that occur more frequently in struggling sessions and might be worth investigating.

By showing this in context, the sidebar can also provide clues as to why people struggle. For example, next pages are labelled according to the referring hyperlink where possible, so authors see what users see, and search terms used in a session provide clues to users' intent.

In future work, we will consider more sophisticated representations of user actions, such as Markov chains [1], which may allow different comparisons between our two sets of users. We are also validating other signals, beyond those described here, which may tell us a visitor is in difficulty. Other records of users' actions may contain further clues, either signals of a struggling session or hints to the cause of confusion, although we are naturally constrained by what we can record at a web server and what users and clients will allow.

We are working with web analysts to check whether these patterns, and the tool in Section 5, do in fact provide useful leads for day-to-day triage and web editing. Although we have not yet carried out a formal evaluation, the tool is in regular use, feedback to date has been good, and analysts have used it to inform changes to their site.

## 8. ACKNOWLEDGEMENTS

We thank the agency's web analysts for their input, as well as access to their log data. Catherine Wise and Brian Jin contributed to the software described in Section 5.

## 9. REFERENCES

[1] J. Borges and M. Levene. Data mining of user navigation patterns. In *Proc. Int. WEBKDD'99 Workshop on Web Usage Analysis and User Profiling*, number 1836 in Lecture Notes in Artificial Intelligence, pages 92–112, 1999.

[2] M. Claypool, P. Le, M. Waseda, and D. Brown. Implicit interest indicators. In *Proc. Intelligent User Interfaces*, pages 33–40, 2001.

[3] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Trans. Info. Systems*, 23(2):147–168, 2005.

[4] B. J. Jansen. *Understanding user-web interactions via web analytics*. Number 6 in Synthesis lectures on information concepts, retrieval, and services. Morgan & Claypool, 2009.

[5] M. Lassila, T. Pääkkönen, P. Arvola, J. Kekäläinen, and M. Junkkari. Unobtrusive mobile browsing behaviour tracking tool. In *Proc. Information Interaction in Context*, pages 278–281, 2012.

[6] C. Liu, J. Gwizdka, and J. Liu. Helping identify when users find useful documents: Examination of query reformulation intervals. In *Proc. Information Interaction in Context*, pages 215–224, 2010.

[7] N. R. Mabroukeh and C. I. Ezeife. A taxonomy of sequential pattern mining algorithms. *ACM Computing Surveys*, 43(1):3:1–3:41, Dec. 2010.

[8] T. Nakayama, H. Kato, and Y. Yamane. Discovering the gap between web site designers' expectations and users' behaviour. *Computer Networks*, 33:811–822, 2000.

[9] F. Silvestri. *Mining Query Logs: Turning Search Usage Data into Knowledge*, volume 4 of *Foundations and Trends in Information Retrieval*. now Publishers, 2010.

[10] K. L. Smith and P. B. Kantor. User adaptation: Good results from poor systems. In *Proc. SIGIR*, pages 147–154, 2008.

[11] M. Spiliopoulou, C. Pohle, and L. C. Faulstich. Improving the effectiveness of a web site with web usage mining. In *Proc. Int. WEBKDD'99 Workshop on Web Usage Analysis and User Profiling*, number 1836 in Lecture Notes in Artificial Intelligence, pages 142–162, 1999.

[12] P. Thomas, A. O'Neill, and C. Paris. Interaction differences in web search and browse logs. In *Proc. Australasian Document Computing Symposium*, Melbourne, 2010.

# Relationship between the nature of the Search Task Types and Query Reformulation Behaviour

*Khamsum Kinley, Dian Tjondronegoro, Helen Partridge and Sylvia Edwards*

Information Systems School, Science and Engineering Faculty
Queensland University of Technology
2 George St Brisbane QLD 4000 Australia

*kkinley@acm.org, {dian, h.partridge, s.edwards}@qut.edu.au*

**Abstract** *Success of query reformulation and relevant information retrieval depends on many factors, such as users' prior knowledge, age, gender, and cognitive styles. One of the important factors that affect a user's query reformulation behaviour is that of the nature of the search tasks. Limited studies have examined the impact of the search task types on query reformulation behaviour while performing Web searches. This paper examines how the nature of the search tasks affects users' query reformulation behaviour during information searching. The paper reports empirical results from a user study in which 50 participants performed a set of three Web search tasks – exploratory, factorial and abstract. Users' interactions with search engines were logged by using a monitoring program. 872 unique search queries were classified into five query types – New, Add, Remove, Replace and Repeat. Users submitted fewer queries for the factual task, which accounted for 26%. They completed a higher number of queries (40% of the total queries) while carrying out the exploratory task. A one-way MANOVA test indicated a significant effect of search task types on users' query reformulation behaviour. In particular, the search task types influenced the manner in which users reformulated the New and Repeat queries.*

**Keywords** Query reformulation behaviour, information behaviour, information retrieval, search task complexity, user studies

## 1 Introduction

Users perform query reformulation and information searches to retrieve relevant information, satisfy their search goals and accomplish their search tasks. However, success of query reformulation and relevant information retrieval depends on many factors that govern users' information searching on the Web, such as task complexity [1], users' prior knowledge [2], age [3], gender [4], information needs [5] and cognitive styles [6, 7]. One of the important factors that affect users' information searching, particularly query reformulation, and information retrieval is that of the nature of the search tasks the user is assigned to. Studies have reported that the amount of time a user spends on searching information on the Web depends on the nature of the search task; thus the nature of the search time affects information searching, including query reformulation.

Gwizdka and Spence [8] reported that the more times searchers spent on a search task, the more Web pages they visited, and the more difficult they faced to assess and access the information. They found that low complexity tasks were characterised by shorter optimal paths (2 to 3 'clicks') and high complexity tasks by longer optimal paths (5 to 6 'clicks'). They also reported that individual differences among Web users affected the relationships between objective task complexity and subjective task difficulty.

Information searchers also tended to use more navigation tools in a general search task that required them to find a fewer pieces of information on a broad topic than they were in a specific task that required locating one specific piece of information that was known to exist on the Web [9].

This paper examines how the nature of the search tasks affects users' query reformulation behaviour during information searching. The paper reports results from a user study in which 50 research participants performed a set of three different assigned Web search tasks.

## 2 Related studies

Task complexity has been identified as having effects on information seeking behaviour by several researchers [e.g., 1, 8, 10-13]. Confronted with a task, the searcher perceives information needs which reflect their interpretation of information requirements, prior knowledge and ability to memorise [11]. The complexity of a task is a central feature in determining its performance and consequent information needs.

Vakkari [10] defines task complexity as a degree of *predeterminability* of task performance. Vakkari reported that the predeterminability of a task could be divided into


*ADCS '12,* December 05 - 06 2012, Dunedin, New Zealand

the predeterminability of its information requirements, process, and outcome. The determinability of a task increases with the increase in the knowledge about its information requirement, process, and outcome. Bilal [14] reported that children's cognitive, physical and affective behaviours are affected by different types of search tasks while searching information on Yahooligans, a Web search engine for children. The study found that children experienced more difficulty with the research task than with the fact-based task. The study also reported that the types of search task influenced children's levels of success.

Choi [15] explored the effects of search task goals, Web search experience, work task stage and topic familiarity on the image searching process. The task goal was defined as the reason or activity that prompts the need to search. The work task stage was a user's assessment of their progress in completing a task. The study reported that most of the search interactions, such as search duration, querying, and navigating, were influenced by contextual factors.

Among the contextual factors, task goals, work task stages and searching experience were found to be the most influential. Users who performed a search for an academic task goal tended to have a longer search session and they also modified their queries frequently. Users with a lower level of search experience were found to spend more time performing searches to employ more querying and navigating tactics and to rate 'usefulness' and 'satisfaction' with search results at a lower rating than those who had a higher level of search experience.

Kim [12] argued that task difficulty depends on an individual searcher's perception, interpretation and judgment of the objective task complexity. She informed that a searcher's background, such as search experience and domain knowledge, specificity and source of information, and search process characteristics, influences the searcher's perception of task difficulty.

Kim and Allen [16] studied the impact of differences in search tasks on Web search activities and outcomes. Their study's findings indicated strong task effects on search activities and outcomes. Different tasks were associated with different levels of search activities and outcomes. Search activities, such as the use of specific search and navigation features, time spent in searching, number of sites viewed, and number of bookmarks created, were found to be influenced by an interaction between cognitive and task variables. For completing a task, searchers spent more time for the subject search task than for the known-item search task and viewed more Web pages for the subject search task than for the known-item search task.

## 2.1 Search Task Classifications

Information researchers tend to categorise information search task attributes from a theoretical prospective, such as "information need" [17], "well-defined/ill-defined" [18], and "task- and fact-oriented" [19]. Borlund and Ingwersen [20] introduced the concept of "simulated work task situation", which discusses the source of information need, the environment of the situation and the problem which has to be solved; and which serves to make the test person understand the objective of the search. They argued that simulated work task situation provides with the context, which ensures "a degree of freedom" to react in relation to individual interpretation of the given situation [20].

Based on the degree of *a priori determinability* (or structuredness) of a task, that the more familiar a task performer is with the task requirement, the less complex the task is perceived, Byström and Järvelin [11] classified task complexity into five categories ranging from an automatic information processing task to a genuine decision task, as illustrated in Figure 1. In automatic information processing tasks, the process, result and types of information used can be described in advance, whereas in genuine decision tasks, none of them can be determined apriori.

Bilal [14] categorised search tasks as fact-based and research. A *fact-based* task is one that required a single, and straightforward answer. A *research* task is one that required the use of critical thinking skills to construct meaning from the relevant information found, and that had multiple facets.

In this research, based on Borlund and Ingwersen's [20] concept of a "simulated work task situation", three search tasks were designed to ensure that these tasks are as close as possible to real world situations (see Section 3.2).

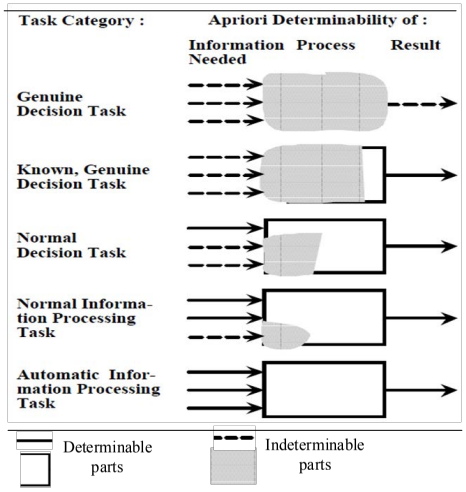



**Figure 1: Task complexity categories (Byström and Järvelin, 1995)**

## 2.2 Research Aims and Questions

While a number of studies have explored the effects of search task complexity on information seeking behaviour in general [e.g. include: 1, 8, 11-13], there is little research conducted on how search task types influence users' query reformulation behaviour.

As both search task types and query reformulations are important components of information searching, this study investigates effects of search task types on the users' query formulation behaviour. This paper also discusses query reformulation classifications adopted in the study. Thus, the research question this research addressed is:

*What are the impacts of search task types on users' query reformulation behaviour?*

# 3 Research Design

## 3.1 Study Participants

Sixty-five (65) responses to the study participation were received either by phone or email return. Of the 65 responses, 50 participants, comprising of students, academics and professional staff from the Queensland University of Technology, were recruited for the study. Efforts were made to include equal number of males and females across different age groups and occupations; this was done following the responses from the prospective participants prior to participation in the study.

## 3.2 Search Task Design

Three types of search tasks were developed: *Factual*, *Exploratory* and *Abstract*. Based on Borlund and Ingwersen's [20] concept of a "simulated work task situation" or scenario, the search tasks were designed to ensure that these tasks are as close as possible to real world situations. The simulated work task situation provides each searcher with the context, which ensures "a degree of freedom" to react in relation to his or her interpretation of the given situation [20]. This approach has been widely used by several researchers in information seeking studies [examples include: 21, 22, 23]. The search tasks designed in this study are listed below:

- Factual: *You have recently moved to Austin, Texas, The U.S., and would like to know the relevant laws passed by the Texas State government regarding child safety while travelling in vehicles. Identify three such rules.*
- Exploratory: *You, with your two friends, are planning a trek for one week in Solukhumbu in Nepal. The trekking will occur next month. You are told that tourists trekking in the place may get high-altitude illness. You decide that you should know more about the place, and the symptoms, seriousness and prevention of high-altitude sickness.*
- Abstract: *You recently heard about the Bermuda Triangle mystery, and you are curious and want to know more about it. So you want to search any relevant information (articles, images and videos) about it and what effect it has on the travellers in the region.*

The *factual* task is a fact-finding search task, such as finding three laws on child safety while travelling in vehicles. The *exploratory* task is more open-ended; there are no specific answers to such task type unlike the factual task. In an *abstract* task, the information need is abstract for which a concrete, direct solution may not exist. The abstract search task is more open-ended than the exploratory task.

## 3.3 Task Complexity

The search tasks were designed with different levels of difficulty and complexity. The main aim of choosing different task complexity was to suit participants with different search experience and skills. It was assumed that the *factual* task has the least complexity, in that the participants were asked to identify any three rules on child restrain while travelling in vehicles in Austin, Texas, which required them to use basic searching skills.

The *exploratory* task was more complex and required a higher level of search experience than for the factual task, in that the participants were asked to search for more information on various topics, such as place (Solukhumbu in Nepal), illness (symptoms of high-altitude illness) and safety measures (preventions of high-altitude illness).

The *abstract* task presented relatively more abstract and complex scenarios compared to the factual and exploratory tasks. The participants needed to organise and structure their search terms carefully by using a more advanced level of search skills and problem solving skills. They needed to find relevant information (that is, articles, images and videos) about the Bermuda Triangle mystery, and its effect on the travellers in the region.

Based on the observation made during the pilot study, in order to break a hierarchical level of task complexity, the exploratory task of second level complexity was issued first, followed by the factual and abstract tasks.

## 3.4 Query Reformulation Classifications

Similar to the previous works in query reformulation type [24-26], we constructed five reformulation categories based on the common and different search terms used in two successive queries: *New*, *Add*, *Remove*, *Replace*, and *Repeat*. Detailed definitions of each of

these queries reformulation classifications with examples are illustrated in Table 1.

## 3.5 Data Collection

An invitation to participate in the study was sent via the university email. This research required a quiet environment, so an individual meeting with the prospective participant for the study participation was scheduled as per the participants' availability. First, each participant was briefed with the participant guidelines and was asked to complete a consent form. As an appreciation of their time and efforts, each study participant received a gift voucher worth AUD $30. Prior to the study, an ethical approval was sought from the University.

User's demographic information was collected using a pre-search questionnaire. Each study participant was assigned with three sets of search tasks. For the Web search task, each participant was provided with a laptop with Internet access. Participants were free to choose whatever search engines they like to use. Although the participants were never stopped while performing their search tasks, it was recommended that they spend between 10 and 15 minutes on each search task.

This study used Web search sessions to investigate each participant's interactions with the search engines. Participants' Web search interactions were captured by using a monitoring program. The output of the program is a video record that can be played and replayed at any time for transcription and analysis.

## 3.6 Data Analysis

The success of a research project depends on the analysis of the data to achieve something interesting and important. A standard search log file format with the following fields, similar to those of Jansen [27], was adopted (see Table 2):

*User Identification*: A unique number used to identify a participant
*Date*: The date of the interaction
*The Time*: The duration of the interaction
*The URL*: The URL of the Web site visited
*Search Terms*: The query terms as entered by the user

This study implemented a quantitative data analysis approach, in which the quantitative data, collected through Web search session logs, were analysed statistically, using SPSS (statistical package for social science). Basic frequency tabulations were used to inform means and standard deviation distribution of the participant demographic and Web search characteristics, such as number of queries and search terms. Advance statistical method, such as multivariate analysis of variance (MANOVA), was performed to investigate to what extent the search task types influence participants' query reformulations.

# 4 Results

## 4.1 Demographic

A total of 50 participants comprising students, academics and professional staff from the Queensland University of Technology participated in the study. Out of 50 participants, 26 were males, accounting for 52 % of

| ID | Date | Time | URL | Search Terms |
|---|---|---|---|---|
| 40 | 03/02/10 | 14:00 | google.com | Bermuda Triangle + effects it has on travellers in the region |
| 40 | 03/02/10 | 14:00 | google.com | Bermuda Triangle + effects on travellers |
| 40 | 03/02/10 | 14:02 | google.com | hypoxia and prevention |
| 40 | 03/02/10 | 14:03 | google.com | hypoxia |

**Table 2: Examples of Web Search Session Logs**

| Type | Description | Query Examples |
|---|---|---|
| *New* | Qi and Qi+1 do not contain any common terms. All new session terms are assigned as a new query. | $Q_i$: "tour"<br>$Q_{i+1}$: "Solukhumbu trek" |
| *Add* | $Q_i$ is a super subset of $Q_{i+1}$, that is, all the terms in Qi are present in $Q_{i+1}$ and $Q_{i+1}$ contains more terms than $Q_i$. | $Q_i$: "Trekking Solukhumbu"<br>$Q_{i+1}$: "Trekking Solukhumbu Nepal" |
| *Replace* | $Q_i$ and$Q_{i+1}$ contain at least one term in common and at least one different term. | $Q_i$: "Tour Nepal"<br>$Q_{i+1}$: "Tour Solukhumbu" |
| *Remove* | $Q_{i+1}$ is a super subset of $Q_i$, that is, all the terms in $Q_{i+1}$ are present in $Q_i$ and $Q_i$ contains more terms than $Q_{i+1}$. | $Q_i$: "Solukhumbu tourist Nepal"<br>$Q_{i+1}$: "tourist Nepal" |
| *Repeat* | $Q_i$ and $Q_{i+1}$ contain exactly the same terms; the order of these terms may be different. | $Q_i$: "trekking Solukhumbu Nepal"<br>$Q_{i+1}$: "Nepal Solukhumbu trekking" |

Note: $Qi_{+1}$ is the succeeding query that follows the query $Q_i$ in the same session

**Table 1: Classifications of query reformulations with examples**

the study sample, and 24 were females (48%). 50% of them were students, 28% staff while 22% of them were both a student and staff at the university.

29 out of 50 participants (58% of the participant population) were aged between 26 and 35 years of age. Three participants were under 20 years of age; 10 participants were aged between 20 and 25 years; 5 between 36 and 45 years of age; two were between 46 and 55 years of age; and one of the participants was over 56 years of age. The study benefited by including participants from different age groups; it was therefore not focused on a particular age group, but rather targeted users of all ages.

## 4.2 Time spent

Total duration of the Web search experiment performed by 50 participants was 26 hours 13 minutes and 50 seconds (rounded to 1574 minutes). Table 3 illustrates time duration for search task. As shown in the table, an average of 10 minutes and 30 seconds was spent on each search task, with a variation of approximately 4 minutes. The minimum searching time spent on a task was 3 minutes and 30 seconds; the maximum time spent was 23 minutes and 25 seconds.

On average, participants took relatively less time to complete the factual task (mean = 9 minutes) compared to the exploratory or the abstract task. We believed that this might be due to that fact that the factual task was assumed to have the least complexity. The participants were required to find only facts that existed: fact-finding tasks are easier to solve because a searcher knows what he or she needs to find. On the contrary, participants spent a longer time on the exploratory task (mean =12 minutes and 47 seconds) because the exploratory task is an open-ended task requiring more time to locate information on the topic.

In the abstract task, participants spent an average of approximately 10 minutes to complete the task. Although the abstract task was assumed to be the most difficult task, on average participants spent less time on completing it than on the exploratory task. It may be due to the fact that the abstract nature of the task provided limited direction for the participants to search on. Overall, the participants spent 40% of their search time on the exploratory task, 29% on the factual task and 31% on the abstract task.

| Task | Mean | SD | Min | Max | Total | % |
|---|---|---|---|---|---|---|
| 1 | 00:12:47 | 00:04:04 | 00:06:05 | 00:23:25 | 10:39:51 | 40% |
| 2 | 00:09:01 | 00:03:42 | 00:03:30 | 00:19:40 | 07:31:18 | 29% |
| 3 | 00:09:39 | 00:03:46 | 00:03:47 | 00:21:35 | 08:02:41 | 31% |
| All | 00:10:30 | 00:04:10 | 00:03:30 | 00:23:25 | 26:13:50 | 100% |

*Note*: Task 1 (Exploratory), 2 (Factual) and 3 (Abstract)

**Table 3: Time Duration for Search Task in hh:mm:ss**

## 4.3 Search queries and terms

During the scenario-based search task experiment, 50 participants submitted 872 unique search queries to complete three search tasks. A *query* is defined as string of terms submitted to a search engine per search session. As illustrated in Table 4, 350 queries were submitted for the exploratory task, 226 for the factual task and 296 for the abstract task.

As illustrated in the table, participants submitted fewer queries for the factual task, which accounted for 26%; the reason being that a fact-finding task requires less searching skills. Participants completed a higher number of queries while completing the exploratory task (40% of the total queries) because it is believed that the exploratory task, being open-ended and requiring searching skills to complete, required more queries to be reformulated. In general, the average number of search queries submitted to complete a task was 5.73.

As shown in Table 5, 50 participants submitted a total of 3613 search terms to complete three search tasks each. A *term* is defined as a series of characters delimited by a white space. The average number of search terms submitted to search engines was 4.14 per query (know as *query length*). Early Web search studies, between 1997 and 2002, reported an average query length between two and three terms [28, 29]. This is something that we intend to explore in detail in future works.

On average, a participant submitted approximately 24 search terms to complete a single search task. However, there was a vast variation in the number of queries being submitted (SD = 21.21), which indicated that participants varied in their query formulating.

In summary, Figure 2 presents the overview of relationships and patterns between search time, search query, and search term across search task types in terms of distributions in percentage (%), with the percentage increasing from the centre towards a vertex of the triangle (for an example the outer line indicates a value of 45%). As illustrated in the figure, participants showed

| Tasks | Mean | SD | Min | Max | Total | % |
|---|---|---|---|---|---|---|
| Exploratory | 7.00 | 3.614 | 1 | 16 | 350 | 40% |
| Factual | 4.52 | 3.570 | 1 | 19 | 226 | 26% |
| Abstract | 5.92 | 3.487 | 1 | 17 | 296 | 34% |
| All Task | 5.73 | 3.61 | 1 | 19 | 872 | 100% |

**Table 4: Frequency of search queries for each task**

| Tasks | Mean | SD | Min | Max | Total | % |
|---|---|---|---|---|---|---|
| Exploratory | 25.46 | 18.10 | 4 | 78 | 1273 | 35% |
| Factual | 26.68 | 26.11 | 4 | 126 | 1334 | 37% |
| Abstract | 20.12 | 19.89 | 3 | 119 | 1006 | 28% |
| All Task | 23.72 | 21.21 | 3 | 126 | 3613 | 100% |

**Table 5: Frequency of search terms for each task type**

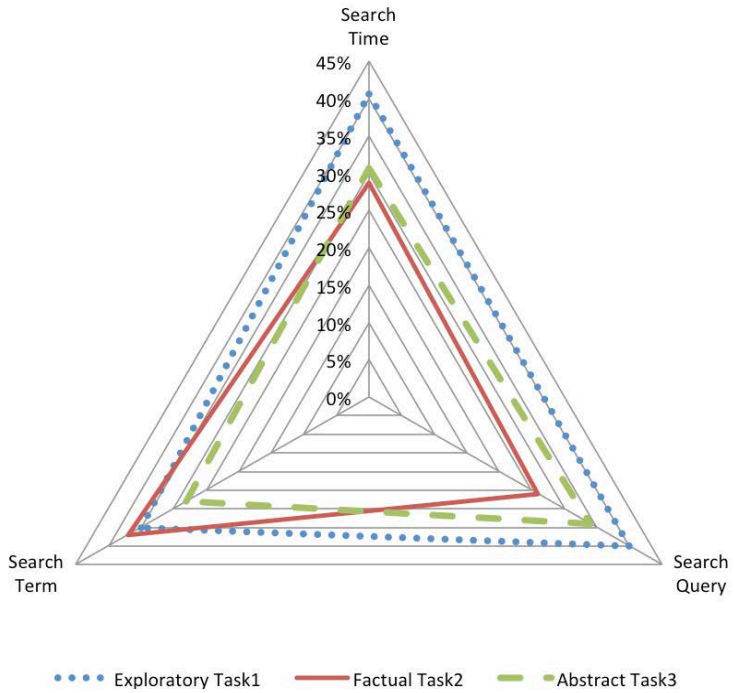



**Figure 2: Search time, search query and search terms by search task types**

similar pattern in terms of search time spent and search query executed across the three search tasks. However, they spent a relatively longer search time and a higher number of search queries for the exploratory task than for the other two tasks; and spent a relatively shorter time and submitted a fewer queries for the factual task.

On the other hand, participants showed contradictory behaviour while submitting search terms. They submitted a relatively higher number of search terms while completing the factual task and the least number of search terms for the abstract task. This indicated that the participants seemed to submit longer queries to search facts on the Web, which may be because they were told what facts to find and they could easily use the given keywords as search terms.

## 4.4 Associations between Search Task Types and Query Reformulation Behaviour

Figure 3 illustrates the overall distributions of the five types of query reformulations across the three search task types. Although all the participants completed all three sets of search tasks, the occurrence of each query types varied across three tasks. In the *exploratory* task, participants executed a higher number of *New* queries; the least was *Repeat* queries. This indicated that while performing exploratory information searching on the Web, participants preferred to search with *New* queries and least with *Repeat*.

Although the number of occurrence of each query type was relatively higher in the exploratory task, the participants seemed to display similar behaviour in the *factual* task. However, in the *abstract* task, participants tended to prefer *Repeat* queries because among the queries, they completed the highest number of *Repeat* queries. There seemed to be two possible reasons for their preference for *Repeat* queries (that is, for repeating search terms):

We believe that the participants might have had limited possible alternative key words because of the abstract nature of the *abstract* search task. Therefore, they might have changed the order and used the same search terms again.

Due to the abstract nature of the task, the participants might have searched the information with the same search query on different search engines, such as Yahoo, Google video or Google images.

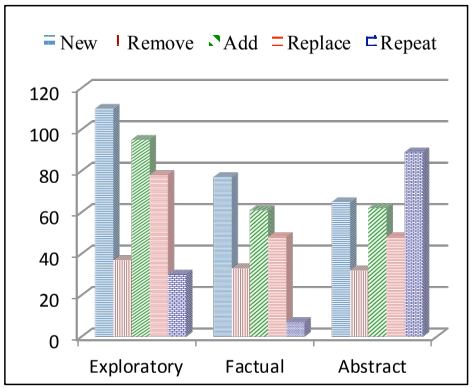



**Figure 3: Distributions of query reformulation types in the three search task types**

A MANOVA test revealed a significant multivariate main effect for search task type, Wilks' $\lambda = .208$, $F(10, 286) = 8.435$, $p < .001$, partial eta squared = .228. Given the significance of the overall test, the univariate main effects were investigated. Significant effects for search task type were obtained for *New*, $F(2, 147) = 12.612$, $p < 0.01$; and *Repeat*, $F(2, 147) = 33.559$, $p < 0.01$. This indicated that the search task types (i.e., exploratory, factual and abstract tasks) influenced the way the participants reformulated *New* and *Repeat* queries.

## 5 Discussions and Conclusion

The study results show that users' query reformulation behaviours are affected by the types of the search tasks. In general, participants tended to formulate more queries to complete the exploratory task and least for the factual task except the *Repeat* queries; a higher number of *Repeat* queries were being formulated for the abstract task than for the exploratory task. Among the queries, participants tended to formulate more *New* queries and lesser *Remove* queries to complete exploratory and factual tasks. However, the number of *Repeat* queries was higher than any other query types for the abstract task.

A one-way MANOVA test results showed that participants' *New* and *Repeat* query reformulations differed across three search tasks. The search task types influenced the manner in which the participants reformulated *New* queries and *Repeat* queries.

The study results would have some implications for search engines designers for the design of query suggestions that are offered to users by search engines during Web searching, and information behaviour (IB) researchers who are concerned about information searchers and their query reformulation behaviours.

Search engines can identify the type of information the user is looking for by capturing the trend of the query reformulations for a particular search, and then provide effective query suggestions accordingly. IB researchers can explore user–Web search interactions through analysis of users' query reformulation behaviour for a particular type of search task. Educators and researchers need to be aware that information searchers' success of retrieving relevant information depends on their query reformulation behaviour, which depends on the nature of the types of search tasks.

## 6 Limitation and Future Work

Participants were assigned with three pre-designed search tasks to complete. Although the assigned search tasks were designed as close as possible to real-world situations, and with a diverse area of topics, the subject motivation was a concern. Some participants were familiar with certain topics, while others were not. These differences in prior knowledge about the subject might have inferred the study's findings. It is believed these limitations can be partially reduced in future works by assigning more search tasks of each type.

Although, in order to break a hierarchical level of task complexity, participants were issued with the exploratory task first followed by the factual and abstract, the time for exploratory could have been inflated with a learning effect and time for the abstract task could have been shortened due to fatigue effect. In future studies, tasks of different types could be issued randomly so that the time difference caused by the fatigue effect is minimal.

This research is also concerned about participants' information needs due to the fact that the search tasks were pre-designed, as these search tasks might have limited the participant's information needs. Their information needs were limited to what was required to perform the assigned search tasks, rather than being given a choice to search their own personal information need.

Future research can explore Web search behaviour in general and query reformulation in particular by asking participants to find solutions to their own identified information problems. The search tasks then can be categorised into different types based on the complexity level.

## References

[1] J. Kim, Task difficulty as a predictor and indicator of web searching interaction, presented at the CHI 2006, 2006.

[2] A. Lazonder, H. Biemans, and I. Wopereis, Differences Between Novice and Experienced Users in Searching Information on the World Wide Web, *Journal of the American Society for Information Science,* vol. 51, pp. 576-581, 2000.

[3] M. Roy and M. T. H. Chi, Gender differences in patterns of searching the web, *Journal of educational computing research,* vol. 29, pp. 335-348, 2003.

[4] A. J. Stronge, W. A. Rogers, and A. D. Fisk, Web-based information search and retrieval: Effects of strategy use and age on search success, *Human Factors: The Journal of the Human Factors and Ergonomics Society,* vol. 48, p. 434, 2006.

[5] T. D. Wilson, On User Studies and Information Needs, *Journal of Documentation,* vol. 37, pp. 3-15, 1981.

[6] N. Ford, B. Eaglestone, A. Madden, and M. Whittle, Web Searching by the "general public": An Individual Differences Perspective, *Journal of Documentation,* vol. 65, pp. 632-667, 2009.

[7] K. Kinley, D. Tjondronegoro, H. Partridge, and S. Edwards, Human-Computer Interaction: The Impact of users' Congitive Styles on Query Reformulation Behaviour during Web searching, in *OZCHI 2012*, November 26–30, 2012, Melbourne, Victoria, Australia, 2012.

[8] J. Gwizdka and I. Spence, What Can Searching Behavior tell us about the Difficulty of Information Tasks? A Study of Web Navigation, in *American Society for Information Science and Technology*, 2006.

[9] K.-S. Kim, Effects of emotion control and task on Web searching behavior, *Information Processing & Management,* vol. 44, pp. 373-385, 2008.

[10] P. Vakkari, Task complexity, Problem Structure and Information Actions Integrating Studies on Information Seeking and Retrieval, *Information Processing and Management,* vol. 35, pp. 819-837, 1999.

[11] K. Byström and K. Järvelin, Task Complexity Affects Information Seeking and Use, *Information Processing and Management,* vol. 31, pp. 191-213, 1995.

[12] J. Kim, Task as a predictable indicator for information seeking behavior on the Web, Ph.D., Rutgers University, New Brunswick, 2006.

[13] J. Kim, Perceived difficulty as a determinant of Web search performance, *Information Research,* vol. 13, 2008.

[14] D. Bilal, Children's use of the Yahooligans! Web search engine: II. Cognitive and physical behaviors on research tasks, *Journal of the American Society for Information Science and Technology,* vol. 52, pp. 118-136, 2001.

[15] Y. Choi, Effects of contextual factors on image searching on the Web, *Journal of the American Society for Information Science and Technology,* vol. 61, pp. 2011-2028, 2010.

[16] K.-S. Kim and B. Allen, Cognitive and Task Influences on Web Searching Behavior, *Journal of the American Society for Information Science and Technology,* vol. 53, pp. 109-119, 2002.

[17] T. D. Wilson, Information Behaviour: An Interdisciplinary Perspective, *Information Processing & Management,* vol. 33, pp. 551-572, 1997.

[18] P. Ingwersen, Cognitive Perspectives of Information Retrieval Interaction: Elements of a Cognitive IR Theory, *Journal of Documentation,* vol. 52, pp. 3-50, 1996.

[19] D. Kelly, X. Yuan, N. Belkin, V. Murdock, and W. Croft, Features of documents relevant to task-and fact-oriented questions, in *11th International Conference on Information and Knowledge Management*, McLean, VA, 2002, pp. 647-650.

[20] P. Borlund and P. Ingwersen, The development of a method for the evaluation of interactive information retrieval systems, *Journal of Documentation,* vol. 53, pp. 225-250, 1997.

[21] P. Borlund, The IIR Evaluation Model: A Framework for Evaluation of Interactive Information Retrieval Systems, *Information Research,* vol. 8, pp. 8-3, 2003.

[22] J. Kim, Describing and predicting information-seeking behavior on the Web, *Journal of the American Society for Information Science and Technology,* vol. 60, pp. 679-693, 2009.

[23] C. Liu, J. Gwizdka, J. Liu, T. Xu, and N. Belkin, Analysis and Evaluation of Query Reformulations in Different Task Types, in *American Society for Information Science and Technology*, 2010.

[24] H. Hoang, T. Nguyen, and A. Tjoa, A Semantic Web-Based Approach for Context-Aware User Query Formulation and Information Retrieval, *International Journal of Information Technology and Web Engineering,* vol. 3, p. 1, 2008.

[25] L. Tseng, D. Tjondronegoro, and A. Spink, Analyzing web multimedia query reformulation behavior, in *14th Australasian Document Computing Symposium*, University of New South Wales, Sydney, NSW, 2009, pp. 118-125.

[26] B. Jansen, D. Booth, and A. Spink, Patterns of query reformulation during Web searching, *Journal of the American Society for Information Science and Technology,* vol. 60, pp. 1358-1371, 2009.

[27] B. J. Jansen, Search Log Analysis: What it is, What's been done, How to do it, *Library and Information Science Research,* vol. 28, pp. 407-432, 2006.

[28] J. F. Hair, W. C. Black, B. J. Babin, R. E. Anderson, and R. L. Tatham, Multivariate Data Analysis, 7th ed. Upper Saddle River, NJ: Prentice Hall, 2010.

[29] B. J. Jansen, A. Spink, and T. Saracevic, Real life, real users, and real needs: a study and analysis of user queries on the web, *Information Processing & Management,* vol. 36, pp. 207-227, 2000.

# Models and Metrics: IR Evaluation as a User Process

Alistair Moffat
Department of Computing and Information Systems
The University of Melbourne
ammoffat@unimelb.edu.au

Falk Scholer
School of Computer Science and Information Technology
RMIT University
falk.scholer@rmit.edu.au

Paul Thomas
ICT Centre, Canberra
CSIRO
paul.thomas@csiro.au

## ABSTRACT

Retrieval system effectiveness can be measured in two quite different ways: by monitoring the behavior of users and gathering data about the ease and accuracy with which they accomplish certain specified information-seeking tasks; or by using numeric effectiveness metrics to score system runs in reference to a set of relevance judgments. The former has the benefit of directly assessing the actual goal of the system, namely the user's ability to complete a search task; whereas the latter approach has the benefit of being quantitative and repeatable. Each given effectiveness metric is an attempt to bridge the gap between these two evaluation approaches, since the implicit belief supporting the use of any particular metric is that user task performance should be correlated with the numeric score provided by the metric. In this work we explore that linkage, considering a range of effectiveness metrics, and the user search behavior that each of them implies. We then examine more complex user models, as a guide to the development of new effectiveness metrics. We conclude by summarizing an experiment that we believe will help establish the strength of the linkage between models and metrics.

## Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and software—*performance evaluation*.

## General Terms

Experimentation, measurement.

## Keywords

Retrieval experiment, evaluation, system measurement.

## 1. OVERVIEW

Information retrieval (IR) systems are measured in two quite different ways. The *efficiency* of an IR system is quantified in terms of CPU, memory and disk resources required, as functions of the volume of data in the system, the rate at which queries must be processed, and the semantics of each query (and hence the query processing modality), and is ultimately demonstrated by experimental measurement of an instrumented implementation. The *effectiveness* of an IR system – the ease with which users of the system can carry out information-seeking tasks and satisfy information needs – is a more subtle concept. We focus on the latter in this paper.

Two approaches to quantifying effectiveness have emerged over the years. The first is the use of what are generically called *user studies*, in which a pool of experimental subjects are given one or more search tasks to carry out, and their actions and behaviors during the prosecution of those tasks are monitored, analyzed, and reported. Provided that the information-seeking tasks used during the experiment are ones that the experimental subjects are able to empathize with, a well-designed user study can provide rich information about all aspects of the system being evaluated, including the interface ("why does it do that when I do this"), robustness ("wow, it crashed again"), and underlying system effectiveness ("it's a bit strange that it didn't put that one on the first page of results").

But user studies are expensive to plan and run, both in terms of actual money, and in terms of time. The planning is costly because of the need to fix all variables and then seek institutional ethics board clearance for a particular experiment, and then recruit subjects; and carrying experiments out is costly because of the need to provide supervision while the subjects are undertaking the specified search tasks. These costs mitigate against continuous user studies in all but the very largest of organizations.

Instead, a second type of effectiveness investigation is common. In a *batch evaluation* (or a *test collection* evaluation), a document collection is compiled; a set of topics or information needs is formulated that can be answered out of that collection; and some or all of the documents in the corpus are *judged* against the topics, to determine whether or not each such document is *relevant* to the specified topic. Those relevance judgments can then be repeatedly used to *score* the outputs of the IR system using a chosen *effectiveness metric* to convert the system outputs into a numeric score. In such an environment, experimental turnaround can be measured in minutes rather than weeks, and a large number of program modifications or parameter settings can be evaluated in a relatively short span of time, and at relatively low cost.

While repeatable, and hence convenient, batch mode experimental evaluation has potential drawbacks:

- Performing comprehensive – or even moderately wide – relevance judgments is a significant initial cost that can only be recouped over a period of time and through multiple uses.
- Effectiveness metrics are typically evaluated over individual queries when used in batch evaluations, whereas a user may


*ADCS'12* December 5-6, 2012, Dunedin, New Zealand.

pose multiple queries as part of a session of activity during an information-seeking activity.

- The metric used might not correlate with the "user experience", meaning that differences in metric scores do not necessarily translate into measurable differences in the user's ability to carry out the desired search task.

Section 6 gives an overview of how previous researchers have addressed these various issues. Our purpose in this paper is explore the relationship between effectiveness metrics and user behavior that is alluded to in the final point, and hence shed further light on the extent to which batch evaluation scores can be argued as having been inspired (or even merely informed) by user behavior.

Section 2 introduces a range of established effectiveness metrics, and for each describes a *user model* that corresponds to the metric. The common thread that links these metrics and models is that they are *static*, and are based on predefined probability distributions. Some of the models are unappealing, in that they do not intuitively resonate with anticipated user behaviors; that reaction can, of course, be interpreted as a suggestion that the metric in question is not particularly appropriate.

In Section 3 we examine *adaptive* models in which the relevance of the documents being inspected comes into play as well as depth in the ranking. Corresponding adaptive effectiveness metrics are introduced for each of the adaptive user models, and their drawbacks considered. Section 4 then asks how models can be compared, and how the choice of an evaluation metric affects the outcomes from comparative IR experiments. Section 5 introduces a new model that better reflects the actions undertaken by typical users, and defines a corresponding effectiveness metric. Section 7 then describes an experiment that might allow confirmation of that user model.

## 2. STATIC USER MODELS

In this section we describe a sequence of user models. Each of them corresponds to an evaluation metric that can be applied post-hoc to runs and relevance judgments, to obtain numeric scores. A *run* is a ranking of documents or snippets, generated by a information retrieval system is response to a query; and a set of judgments (sometimes called a *qrels* file) is a record of which documents have been judged relevant for that query. Note that there is no requirement that relevance must be binary, and throughout our discussion it is assumed that relevance is (possibly quantized values selected from) a continuous scale $0 \le r \le 1$, with $r = 0$ meaning "no relevance at all" and $r = 1$ meaning "highly relevant".

Another way of thinking about relevance is that it is the *utility* the user gains if or when they view that document in the ranking. The goal of the user is to gain utility at the highest possible rate, where the unit of cost expended is a document viewing. Hence, a system that more successfully places highly relevant documents amongst the first ones viewed by the user will be a more effective system; and this is what an effectiveness metric should reflect.

### *Precision*

In this simplest scenario, imagine a user who without variation inspects the first $k$ proposed answers in the result listing; and, once they have done so, makes use of the subset of them that are relevant. That is, the user performs $k$ units of work, and gains some utility as a result. Taking $\mathsf{Rel}(k) = \sum_{i=1}^{k} r_i$ as the sum of the relevance scores of those first $k$ documents, where $r_i$ is the relevance score of the document in the $i$ th position in the ranking, gives a measure of that utility, and hence $\mathsf{Rel}(k)/k$ is the rate at which utility has been attained. If the relevance judgments are binary, then $\mathsf{Rel}(k)$ is the number of relevant documents, and $\mathsf{Rel}(k)/k$, is just the standard definition of *precision at depth* $k$, or $\mathsf{P}@k$. That is, the metric $\mathsf{P}@k$ has as a corresponding model that the user always even-handedly inspects exactly $k$ documents in the result listing of each and every query that they pose to the retrieval system.

It is also possible to interpret the previous scenario in a probabilistic sense, and infer a uniform probability distribution over the $k$ documents and note that $\mathsf{P}@k$ is the expected relevance that accrues from a user selecting and inspecting a single random document according to that distribution, spending one unit of work as they do:

$$\mathsf{W}_{\mathsf{Prec}}(i) = \begin{cases} 1/k & \text{when } 1 \le i \le k \\ 0 & \text{otherwise.} \end{cases}$$

With this definition, the effectiveness score computed for a ranking can be thought of as being the inner-product of a pre-defined weighting vector and a relevance vector $r = \langle r_i \rangle$. That is,

$$\mathsf{P}@k = \sum_{i=1}^{k} r_i \cdot \mathsf{W}_{\mathsf{Prec}}(i) = \sum_{i=1}^{\infty} r_i \cdot \mathsf{W}_{\mathsf{Prec}}(i)\,,$$

where the sum can be extended to infinity because of the zeros in $\mathsf{W}_{\mathsf{Prec}}(i)$. With this formulation in place, any other probability distribution over the integers $1 \ldots \infty$ can also be used as the basis for a *weighted precision* effectiveness metric.

### *Scaled Discounted Cumulative Gain*

Järvelin and Kekäläinen [8] observe that top-weightedness of evaluation metrics is desirable, writing "... the greater the ranked position of a relevant document ... the less likely it is that the user will ever examine it", and describe an inner-product metric they call *discounted cumulative gain*, or $\mathsf{DCG}@k$. In their description, Järvelin and Kekäläinen [8] make use of a vector of weights that in fact is not a probability distribution, multiplying the relevance of the $i$ th item in the ranking by $1/\max\{1, \log_b i\}$; that initial formulation has since evolved in use to become $1/\log_2(i+1)$. Note that the inverse logarithmic sequence is not bounded, and that raw $\mathsf{DCG}$ effectiveness scores have no upper limit. To generate a probability distribution, and hence ensure that effectiveness scores are in the range $[0, 1]$, the evaluation depth $k$ must be fixed, and a truncated and scaled weight vector employed:

$$\mathsf{W}_{\mathsf{SDCG}}(i) = \begin{cases} (1/S(k)) \cdot (1/\log_2(i+1)) & \text{when } 1 \le i \le k \\ 0 & \text{otherwise.} \end{cases}$$

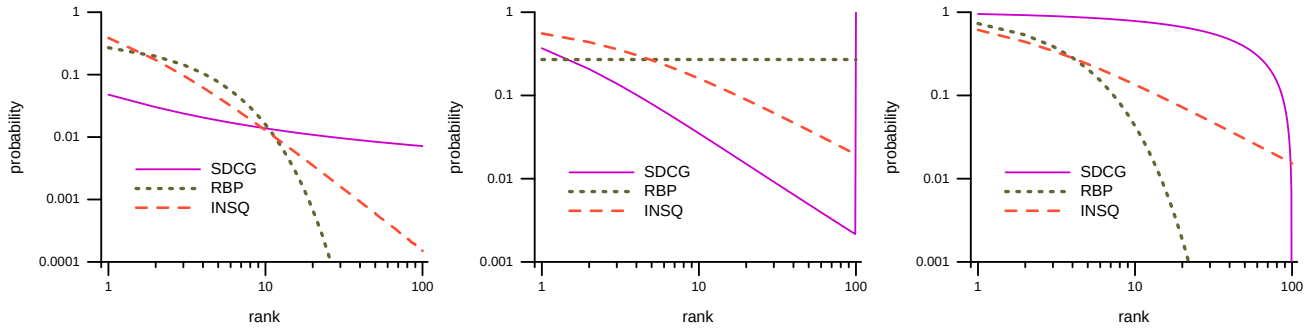
where

$$S(k) = \sum_{i=1}^{k} \frac{1}{\log_2(i+1)}$$

is the necessary scaling constant. We denote the resultant effectiveness metric as *scaled discounted cumulative gain*,

$$\mathsf{SDCG}@k = \sum_{i=1}^{\infty} r_i \cdot \mathsf{W}_{\mathsf{SDCG}}(i)\,.$$

The corresponding user model represents users as having determined in advance that they will examine exactly $k$ items in the result listing, and within that set of $k$ documents, will be somewhat biased in favor of those near the top of the ranking, but also with a non-trivial interest in all of the answers through to the $k$ th. Figure 1a shows the distribution that arises when $k = 100$, with item weight plotted a function of depth in the ranking. As can be seen, $\mathsf{SDCG}@100$ is somewhat top-weighted. But the bias is relatively small, and the document in the first position in the ranking is only seven times more likely to be examined than the document in position 100. Put another way, the model defined by $\mathsf{SDCG}@100$




(a) Weight $\mathsf{W}_\mathsf{M}()$ (b) Halting probability $\mathsf{H}_\mathsf{M}() = 1 - \mathsf{C}_\mathsf{M}()$ (c) Residual $\mathsf{R}_\mathsf{M}()$

**Figure 1:** Weights, halting probabilities, and residuals as a function of rank, for weighted-precision metrics SDCG@100, RBP with $p = 0.73$, and INSQ. All scales are logarithmic.

suggests that around one in seven searches reaches depth 100, but that no searches ever go to position 101 and beyond.

### *Rank-Biased Precision*

As an alternative way of addressing the non-convergence of the inverse logarithmic sequence, Moffat and Zobel [10] suggest the use of the infinite *geometric distribution* to construct a metric they call *rank-biased precision*, or RBP, specified by:

$$\mathsf{W}_{\mathsf{RPB}}(i) = (1-p)p^{i-1}\,,$$

where $p$ is a *persistence* parameter. Moffat and Zobel [10] also describe the user model that accompanies this probability distribution, supposing that the user always views the first answer in the ranking, and, having viewed the document at rank $i$, views the document at rank $i+1$ with a fixed conditional probability $p$. On average, a user will thus examine $1/(1-p)$ documents in the ranking.

A benefit of the use of the geometric distribution is that it converges, and hence the RBP@$k$ metric is monotonic as the depth of evaluation $k$ is increased. Neither P@$k$ nor SDCG@$k$ have this property. In both of them, as the depth of evaluation increases from $k$ to $k'$, scores for P@$k$ and SDCG@$k$ do not provide lower bounds for scores P@$k'$ and SDCG@$k'$. The fact that the sequence of weights used in RBP converges also means that at any given depth of evaluation an upper bound on the eventual metric score can also be computed, based on the sum of the tail of the distribution [10]. Hence, it is possible to drop the "@$k$" part of the metric and refer to it as RBP; as a result, all of P, SDCG, and RBP are metrics with a single parameter each.

### *Inverse Squares*

Any other infinite convergent distributions can also be employed, suitably normalized so that the sum is 1.0. One such alternative is given by inverse squares of ranks:

$$\mathsf{W}_{\mathsf{INSQ}}(i) = \frac{1}{S}\cdot\frac{1}{(i+1)^2}\,, \tag{1}$$

with

$$S = \frac{\pi^2}{6} - 1 \approx 0.6449\,,$$

which is a probability distribution because of the properties of the Riemann function, $\zeta(2) = \sum_{i=1}^{\infty}(1/i^2) = \pi^2/6$. Figure 1a includes the infinite weighting functions $\mathsf{W}_{\mathsf{RBP}}(i)$, plotted with $p = 0.73$, and $\mathsf{W}_{\mathsf{INSQ}}(i)$. Both are more heavily top-weighted than is SDCG.

### *Halting and continuing*

In these metrics it is assumed that the user scans the items in the result listing from top to bottom, and stops at some point and abandons that query. That assumption allows another probability distribution to be used to characterize each of the models [3]: the probability (according to metric M) that each item in the ranking is the last one inspected, computed as:

$$\mathsf{L}_\mathsf{M}(i) = \frac{\mathsf{W}_\mathsf{M}(i) - \mathsf{W}_\mathsf{M}(i+1)}{\mathsf{W}_\mathsf{M}(1)}\,,$$

which describes a probability distribution because the sequence of weights is decreasing, and because $\mathsf{W}_\mathsf{M}(1)$ is the largest weight. For example, $\mathsf{L}_{\mathsf{Prec@100}}(100)$ is 1.0 and $\mathsf{L}_{\mathsf{Prec@100}}(i) = 0.0$ at all other points $i$. The 100th item in the ranking is always the last one inspected in this metric.

Another set of values can be derived from weight distribution $\mathsf{W}_\mathsf{M}(i)$ associated with metric M – the conditional probability of viewing the $i+1$ th item in the ranking, given that the $i$th has just been examined:

$$\mathsf{C}_\mathsf{M}(i) = \frac{\mathsf{W}_\mathsf{M}(i+1)}{\mathsf{W}_\mathsf{M}(i)}\,.$$

For example, in the user model associated with RBP, the conditional probability of viewing the $i+1$ st item in the result listing, given that the $i$th item has just been viewed, is always $p$. Figure 1b plots $\mathsf{H}_\mathsf{M}(i) = 1 - \mathsf{C}_\mathsf{M}(i)$, the conditional probability at depth $i$ of halting the search at that point. Because SDCG@100 uses a truncated distribution, the conditional halting probability is 1.0 at depth 100. On the other hand, the weight, last, and halting probabilities for RBP and INSQ are smooth distributions.

### *Residuals*

If any one of the four distribution $\mathsf{W}_\mathsf{M}()$, $\mathsf{L}_\mathsf{M}()$, $\mathsf{C}_\mathsf{M}()$, or $\mathsf{H}_\mathsf{M}()$ is provided for some metric M, the other three can be inferred. In addition, note that, by construction,

$$\mathsf{R}_\mathsf{M}(k) = \sum_{i=k+1}^{\infty}\mathsf{W}_\mathsf{M}(i) = \prod_{i=1}^{k}\mathsf{C}_\mathsf{M}(i)\,.$$

That is, the *residual* – the sum of the weight of the non-included tail at depths $k+1$ and beyond for metric M – is given by the product of the first $k$ conditional continuation probabilities. The residual represents the score uncertainly that arises when relevance assessments $r_i$ are only known for the first $k$ elements in the ranking. For example, $\mathsf{R}_{\mathsf{SDCG@100}}(100) = 0.0$, by construction; whereas $\mathsf{R}_{\mathsf{INSQ}}(100) \approx 0.0151$. With RBP and $p = 0.73$ the same level

of residual is achieved earlier, because of the steeper drop-off in the weight distribution compared to INSQ. With larger values of $p$ – representing more persistent searching – that relationship alters. Figure 1c plots residual functions $\mathsf{R}(i)$ for SDCG, RBP with $p = 0.73$, and INSQ.

### *Are static user models realistic?*

All of these four static metrics – P@$k$, SDCG@$k$, RBP, and INSQ – can be criticized. For example, P@$k$ is not top-weighted, and SDCG@$k$ only moderately so. Moreover, the truncated weight distributions used by P and SDCG prohibit user access beyond depth $k$, an aspect of their structure that is unlikely to be reflected in user performance. Similarly, RBP can be criticized because the conditional halting probability is constant at all depths, whereas it seems reasonable to suppose that a user who reaches depth 42 is less likely to stop at that point than is a user who has just examined the 2 nd document in the ranking; and a user at depth 92 is even less likely to not look at another document. That is, it seems natural for halting probabilities $\mathsf{H}_{\mathsf{M}}(i)$ to decrease as a function of depth.

In addition, when values of $p$ below 0.9 or so are used, RBP underweights the deep part of the distribution (with $p = 0.73$, the document at depth 100 has a weight of just $6 \times 10^{-15}$); but when $p$ is closer to 1, the distribution probably underweights the top part of the distribution (at $p = 0.95$, $\mathsf{W}_{\mathsf{RBP}}(1) = 0.05$).

The fixed model associated with INSQ rectifies many of these deficiencies – it avoids truncation, the halting probability decreases with depth, and it assigns plausible weights at both the top of the ranking ($\mathsf{W}_{\mathsf{INSQ}}(1) = 0.388$, $\mathsf{W}_{\mathsf{INSQ}}(2) = 0.172$, and $\mathsf{W}_{\mathsf{INSQ}}(3) = 0.097$) as well as further down ($\mathsf{W}_{\mathsf{INSQ}}(100) = 1.5 \times 10^{-4}$). It could also be parameterized through the use of a different power than 2, or a different additive constant than 1, to shift the three curves plotted in Figure 1.

But the real failing of static metrics is that, in terms of a user model, none of them take into account what it is that the user is experiencing as they step down the ranking. That is, static metrics completely ignore the fact that as the user examines documents they either make progress towards their search goal or they do not, and their internal assessment of the task they are working on must be evolving. Indeed, unless the user is completely agnostic as to the outcome of their search session, their behavior must of necessity differ as they do, or do not, get closer to answering the question they sought to answer. For example, they will terminate their search as soon as (or not long after) their information need has been satisfied, regardless of what they have done up until that point. This is a critical failing that has been noted by a number of authors (see, for example, Chapelle et al. [5]).

## 3. ADAPTIVE USER MODELS

We now consider methods in which the user model is sensitive to the relevance of the documents being examined.

### *Reciprocal Rank*

Using the definitions already established, reciprocal rank, or RR, is given by:

$$\mathsf{L}_{\mathsf{RR}}(i) = \begin{cases} 1 & \text{if } i = \arg\min_j \{r_j \mid r_j = 1\} \\ 0 & \text{otherwise}. \end{cases}$$

The corresponding model is that the user inspects all documents in the ranking down to, and including, the first relevant one; they always end their search at the first relevant document encountered. The score is again a "rate of utility gained per unit of effort spent", since one unit of relevance is gained, out of the $\arg\min_j \{r_j \mid r_j = 1\}$ documents examined during a sequential search.

### *Average Precision*

Reciprocal rank requires knowledge in the ranked list of the position of the first-appearing relevant document; *average precision*, or AP, can be viewed as being a generalization of RR in which knowledge is required of the positions of *all* of the relevant documents. Like RR, it is most conveniently expressed in terms of the last document probability $\mathsf{L}(i)$,

$$\mathsf{L}_{\mathsf{AP}}(i) = \begin{cases} r_i/R & \text{if } R > 0 \\ 0.0 & \text{otherwise}. \end{cases}$$

where $R = \mathsf{Rel}(N) = \sum_{i=1}^{N} r_i$ is the total sum of the relevance for that query over all of the $N$ documents in the collection being ranked. The other distributions, $\mathsf{W}_{\mathsf{AP}}(i)$ and $\mathsf{C}_{\mathsf{AP}}(i)$ can be derived from $\mathsf{L}_{\mathsf{AP}}(i)$, as discussed in the previous section.

The corresponding user model is one in which a user selects at random one of the relevant documents (in the case of multi-grade relevance, with the selection biased by the degree of relevance) and then examines every document down to and including that one in the result listing [11]. The score assigned by the metric is again an expected rate at which utility is gained.

### *Are adaptive user models realistic?*

As was the case with the static models, questions are quick to arise when the plausibility of the user models is considered. The model for RR requires that users scan through to the first relevant document in the ranking, regardless of how deeply it appears; the model for AP is even more contrived, in that it suggests that a user somehow intuits how many relevant documents there are in the ranking, and then scans past (on average) half of them before stopping, regardless of how far through the ranking that might take them, and regardless of how many answers they are interested in finding.

In terms of calculating scores, AP has the additional drawback that a value for the metric cannot be computed until $R$ is known (or somehow approximated), which requires rather more work than (say) just judging the first $k$ documents in the ranking, as is required for P@$k$ and SDCG@$k$; or scanning the ranking until a relevant document is encountered, as is the case for RR. Another area for concern is that neither RR nor AP are defined if there are no relevant documents for the query. Despite these concerns, AP and RR are widely used in retrieval experimentation. Other adaptive metrics (for example, the recently-proposed ERR expected reciprocal rank metric [5]) have yet to gain traction.

## 4. COMPARING METRICS

Having compared the various metrics based on philosophical grounds, it is also of interest to determine if they can be compared empirically in some way. One desirable attribute of a metric is the ability to differentiate systems, since we are typically interested in determining which system is obtaining the highest scores.[1] Hence, one way of evaluating effectiveness metrics is to apply them to system runs generated in shared-task experimental regimes, and examine their ability to differentiate between the systems that contributed to the experiment in a statistically significant manner.

For example, in the TREC-10 Web Track a total of 97 system runs were submitted for evaluation, meaning that there are 4,656

[1]Note, however, that this is a somewhat circular argument, since we are only interested in separating systems if the metric is capturing some essence of the systems that is believed to be important to usability and usefulness. A metric shouldn't be chosen purely because it provides consistent system separations. The name of the system gives completely unambiguous system separations, but is clearly not an interesting reflection of retrieval performance and shouldn't be taken to be an effectiveness metric.

| | rr | insq | p10 | rbp73 | sdcg10 | p100 | rbp95 | sdcg100 | ap |
|---|---|---|---|---|---|---|---|---|---|
| rr | **55.6** | 53.8 | 51.2 | 52.9 | 52.4 | 49.1 | 51.6 | 51.5 | 50.2 |
| insq | 0.0 | **63.8** | 59.4 | 62.4 | 61.5 | 56.4 | 60.1 | 59.9 | 58.1 |
| p10 | 0.0 | 0.0 | **64.5** | 61.1 | 62.7 | 59.1 | 63.3 | 62.2 | 60.4 |
| rbp73 | 0.0 | 0.0 | 0.0 | **64.7** | 63.2 | 57.6 | 61.7 | 61.2 | 59.4 |
| sdcg10 | 0.0 | 0.0 | 0.0 | 0.0 | **64.9** | 58.5 | 62.7 | 62.0 | 60.0 |
| p100 | 0.2 | 0.0 | 0.0 | 0.0 | 0.0 | **68.8** | 63.8 | 66.9 | 64.9 |
| rbp95 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | **69.1** | 67.2 | 64.3 |
| sdcg100 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | **71.0** | 66.3 |
| ap | 0.3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | **72.5** |

(a) Significance agreements and disagreements, $p = 0.05$

| | rr | insq | p10 | rbp73 | sdcg10 | p100 | rbp95 | sdcg100 | ap |
|---|---|---|---|---|---|---|---|---|---|
| rr | — | 85.3 | 77.7 | 81.8 | 80.2 | 65.9 | 71.4 | 68.0 | 62.1 |
| insq | 90.0 | — | 86.8 | 94.8 | 92.0 | 70.9 | 81.1 | 77.2 | 68.4 |
| p10 | 85.2 | 92.6 | — | 90.3 | 94.2 | 77.4 | 89.5 | 83.1 | 74.3 |
| rbp73 | 87.9 | 97.1 | 94.7 | — | 95.5 | 72.5 | 84.1 | 79.3 | 70.7 |
| sdcg10 | 87.0 | 95.6 | 96.8 | 97.5 | — | 74.8 | 86.8 | 81.3 | 72.1 |
| p100 | 79.2 | 85.2 | 88.7 | 86.3 | 87.5 | — | 83.4 | 90.0 | 80.4 |
| rbp95 | 82.8 | 90.4 | 94.8 | 92.2 | 93.5 | 92.5 | — | 90.5 | 77.7 |
| sdcg100 | 81.4 | 88.9 | 91.9 | 90.2 | 91.2 | 95.7 | 95.9 | — | 81.1 |
| ap | 78.7 | 85.3 | 88.2 | 86.6 | 87.3 | 91.9 | 90.8 | 92.5 | — |

(b) Class agreements, $p = 0.05$

**Table 1:** TREC-8 Adhoc Track (1999): 129 systems and 8,256 system pairs, evaluated over 50 topics. Details are explained in the text.

| | rr | insq | p10 | rbp73 | sdcg10 | p100 | rbp95 | sdcg100 | ap |
|---|---|---|---|---|---|---|---|---|---|
| rr | **49.9** | 46.6 | 43.2 | 45.9 | 45.3 | 35.7 | 42.2 | 40.6 | 40.1 |
| insq | 0.0 | **55.5** | 50.7 | 54.2 | 53.2 | 41.3 | 49.8 | 47.4 | 46.3 |
| p10 | 0.0 | 0.0 | **58.8** | 53.5 | 55.6 | 44.5 | 54.3 | 50.8 | 49.3 |
| rbp73 | 0.0 | 0.0 | 0.0 | **58.0** | 56.2 | 42.8 | 52.1 | 49.2 | 47.9 |
| sdcg10 | 0.0 | 0.0 | 0.0 | 0.0 | **58.6** | 44.1 | 53.4 | 50.5 | 48.8 |
| p100 | 0.5 | 0.2 | 0.2 | 0.2 | 0.2 | **54.1** | 49.8 | 52.3 | 49.3 |
| rbp95 | 0.1 | 0.0 | 0.0 | 0.1 | 0.1 | 0.0 | **62.7** | 56.6 | 55.1 |
| sdcg100 | 0.2 | 0.1 | 0.1 | 0.1 | 0.1 | 0.0 | 0.0 | **60.0** | 53.2 |
| ap | 0.5 | 0.1 | 0.1 | 0.1 | 0.1 | 0.0 | 0.0 | 0.0 | **62.5** |

(a) Significance agreements and disagreements, $p = 0.05$

| | rr | insq | p10 | rbp73 | sdcg10 | p100 | rbp95 | sdcg100 | ap |
|---|---|---|---|---|---|---|---|---|---|
| rr | — | 87.1 | 75.6 | 82.7 | 80.4 | 67.1 | 68.2 | 68.6 | 64.4 |
| insq | 88.4 | — | 84.9 | 94.1 | 91.0 | 70.6 | 77.5 | 75.8 | 69.3 |
| p10 | 79.5 | 88.7 | — | 88.5 | 92.6 | 73.0 | 83.8 | 79.1 | 71.4 |
| rbp73 | 85.2 | 95.5 | 91.8 | — | 95.0 | 70.5 | 79.4 | 76.4 | 69.3 |
| sdcg10 | 83.5 | 93.3 | 94.8 | 96.4 | — | 72.4 | 81.6 | 78.8 | 70.2 |
| p100 | 69.3 | 75.6 | 79.1 | 76.8 | 78.5 | — | 79.3 | 88.9 | 78.4 |
| rbp95 | 75.2 | 84.4 | 89.5 | 86.5 | 88.0 | 85.3 | — | 87.7 | 79.8 |
| sdcg100 | 74.2 | 82.2 | 85.7 | 83.5 | 85.4 | 91.7 | 92.3 | — | 79.3 |
| ap | 72.0 | 78.6 | 81.4 | 79.7 | 80.6 | 84.6 | 87.9 | 86.9 | — |

(b) Class agreements, $p = 0.05$

**Table 2:** TREC-10 Web Track (2001): 97 systems, and 4,656 system pairs, evaluated over 50 topics. Details are explained in the text.

"system S1 versus system S2" pairwise system comparisons that can be considered. In addition, if metric A and metric B are both used to score systems S1 and S2, then a total of five different outcomes are possible in terms of confidence indicators from a test for statistical significance, categorized as follows:

SSA Active agreements, where metric M1 and M2 both provide evidence that system S1 is significantly superior to S2, or vice versa on systems;

SSD Active disagreements, where metric M1 says that S1 is significantly better than S2, but metric M2 says that S2 is significantly better than S1, or vice versa on systems;

SN Passive disagreements, where metric M1 provides evidence that system S1 is significantly better than S2 (or vice versa on systems), but metric M2 does not provide evidence in support of the same claim;

NS Passive disagreements, where metric M2 provides evidence that system S1 is significantly better than S2 (or vice versa on systems), but metric M1 does not provide evidence in support of the same claim;

NN Passive agreements, where metric M1 fails to provide sufficient evidence that system S1 is significantly better than S2, and so does metric M2;

Tables 1 and 2 shows the result of such a comparison using the documents, runs, and judgments associated with the TREC-8 Adhoc Track (newspaper articles) and the TREC-10 Web Track (web documents). Similar results were obtained on TREC-9 Adhoc Track and TREC-9 Web Track data; those outcomes are omitted.

In part (a) of each table, the diagonal numbers (in bold) show the discriminative power of the metric in question, calculated as the proportion of all system pairs that are deemed to be significantly different, for that metric. As has been noted by other authors, there is a clear trend whereby metrics that take longer sections of the ranked search results lists into account are able to identify a larger fraction of statistically significant differences between systems. This holds for both collection types. The numbers above the diagonal in part (a) of the tables show the percentage of system pairs for which both metrics agree that one system is significantly superior to another, and both agree which is the better system (category SSA). The numbers below the diagonal show the number of systems for which both metrics show a significant difference between systems, but disagree as to which of the two systems is better (category SSD). Fortunately, this number is generally very small for the web collection, and zero for most pairs of metrics when evaluating the newswire collection.

Part (b) of each table shows two types of class agreement, again as percentages: $2\mathsf{SSA}/(2\mathsf{SSA}+\mathsf{SN}+\mathsf{NS})$ above the diagonal, and $2\mathsf{NN}/(2\mathsf{NN}+\mathsf{SN}+\mathsf{NS})$ below the diagonal. Numbers above the diagonal show the percentage agreement when both metrics report significant differences, while numbers below the diagonal show the percentage agreement where no significant difference between runs is reported.

These agreement scores represent the outcomes of "real" batch-mode IR experiments. In particular, the numbers on class agreement for significance (above the diagonal in part (b) of each table) show cases where a researcher would have concluded that the performance of one algorithm is substantially better than another, with a real effect that was highly unlikely to have been due to chance variation (at the 95% confidence level). For example, consider the column for AP, perhaps the most widely reported effectiveness metric in IR studies. The agreement between AP and other metrics ranges from 62% to 81% across the two collections. Hence, a researcher who conducted the same IR experiment, but measured the outcomes using a metric other than AP, would have rejected the results as being not significant (and hence uninteresting) around 19% to 38% of the time.

The gap between metric behaviors is problematic because, as discussed, there is currently no principled way in which to choose one evaluation metric over another. While there may be broad agreement in the community that certain metrics are more appropriate for certain task types (for example, RR is considered more appropriate for navigational searches than AP), the real differences between metrics are not well understood. The choice between AP, RBP (with a high $p$ value), and SDCG (with a high $k$ value) for

| Initial expectation | Answer occurrence observed after query issued | | |
|---|---|---|---|
| | No answers | Some answers | Many answers |
| Few answers (navigational) | Quickly dissatisfied, early reformulation | Possibly satisfied without needing reformulation | Satisfied quickly, no reformulation |
| Many answers (informational) | Dissatisfied, but will have looked down ranking before reformulating | Partially satisfied, will reformulate after looking down ranking | May be satisfied after first query, if not, will reformulate |

**Table 3:** Hypothesized user search behavior, as influenced by two factors: the anticipated number of answers required, and the extent to which relevant documents are identified while searching. If the query is reformulated, the user's expectation in the followup query will be adjusted to account for relevance carried forward.

evaluating informational searches is largely arbitrary, and yet can lead to different experimental conclusions. It is therefore vital that a better understanding of metrics be developed, and one particular aspect that can help to determine the suitability of a metric is how closely they match real searcher behavior.

# 5. USER-INSPIRED ADAPTATION

Having argued that existing static and adaptive metrics are flawed in various ways, an obvious question is whether any metric exists that meets all of the design goals that were advocated in Sections 2 and 3. Such a metric should:

1. Be computable based on properties of a ranking, without requiring properties of the whole collection to be established.
2. Be top-weighted, but retain non-negligible weight $\mathsf{W}_{\mathsf{M}}(i)$ at ranks of $i \geq 100$ and beyond, and be, as far as possible, a smoothly varying function of $i$, without being truncated.
3. Have a conditional halting probability $\mathsf{H}_{\mathsf{M}}(i)$ that decreases with depth.
4. Adapt to relevant documents in the answer ranking.
5. Be parameterized in accordance with the user's initial rationale for undertaking the search.

To motivate the fifth of these goals, note that it is now accepted that there are different types of information-seeking tasks, including *navigational* interactions, where the purpose is to identify a single answer; and *informational* interactions, in which the user may be seeking to synthesize a new document by drawing on a range of a dozen or more existing ones. Legal and medical search are extreme examples of the latter; and in those disciplines a user commencing an information-seeking task might anticipate spending many hours carrying out a sequence of searches, with a view to identifying scores or even hundreds of relevant documents. That is, we believe that users commence different types of task with different expectations as to how many answers they anticipate finding, and that this expectation affects their search behavior.

To develop a user model we suggest that *the conditional probability of a user continuing their search having reached some depth $i$ in the ranking is a combination of three factors: the depth in the ranking that has been reached; the anticipated number of answers; and the number of answers that have been identified so far through to that depth.* That is, we hypothesize that the conditional continuation function $\mathsf{C}_{\mathsf{M}}(i)$ is positively related to $T$, the anticipated number of answers, and inversely correlated with $\mathsf{Rel}(i) = \sum_{j=1}^{i} r_i$, the amount of relevance identified down to depth $i$ in the ranking.

For example, consider a user undertaking an informational query, with an initial (unvoiced and unexpressed) anticipation of finding perhaps 10 documents. If the first few documents in the ranking are not relevant, the user remains likely to continue looking down the ranking – after all, they were never going to stop after just one document. Alternatively, if relevant documents are encountered early, the user's mental state changes, and they are now (still unvoiced and unexpressed) anticipating finding further answers relatively quickly, after the early wins already attained.

A user that issues a navigational query has quite different behavior. They commence with the expectation that one answer will suffice, and are likely to stop as soon as a relevant document is found. Moreover, they are relatively impatient for that to happen. If the first and second documents are not relevant, they might reformulate even before looking at the third. Table 3 outlines the hypothesized mixture of behaviors.

To formalize these ideas, suppose that at the moment a user issues a query they anticipate needing $T$ relevant documents. To capture their subsequent behavior, we envisage an effectiveness metric that has *two* components – a depth-based *background* conditional continuation probability $\mathsf{C}_{\mathsf{M}}(i)$ that models (as a function of depth) the user's actions in the absence of any relevant documents appearing in the ranking; and a *discounting* modification that is used to adjust that probability as $\mathsf{Rel}(i)$ increases relative to $T$. Together they yield an *adjusted continuation probability* $\mathsf{C}'_{\mathsf{M}}(T,i)$ that incorporates the required influences.

There are, of course, many options that suit these requirements. We now propose one arrangement that meets the hurdle of being "reasonable", even though we are not in a position to provide any evidence that it is "right".[2] As an underlying background model for user activity in the absence of any relevant documents, we parameterize the INSQ metric by adjusting it for $T$, the anticipated number of documents:

$$\mathsf{W}_{\mathsf{INSQ}}(T,i) = \frac{1}{S_{2T-1}} \cdot \frac{1}{(i+2T-1)^2}\,, \tag{2}$$

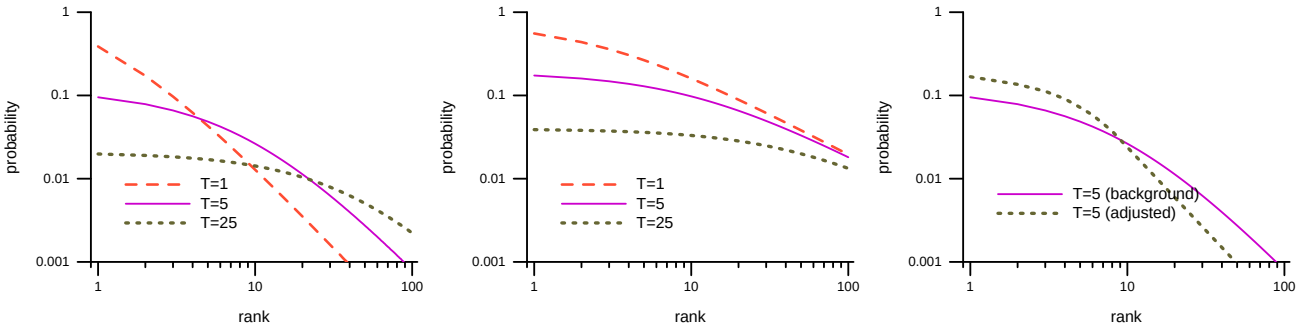
where $S_k = (\pi^2/6) - (\sum_{i=1}^{k} 1/i^2)$ is the normalization constant, and hence that

$$\mathsf{C}_{\mathsf{INSQ}}(T,i) = \frac{(i+2T-1)^2}{(i+2T)^2}\,.$$

When $T > 1$, this has the effect of "flattening" the $\mathsf{W}_{\mathsf{INSQ}}(i)$ curves, decreasing the weights when $i$ is small, and increasing them when $i$ is large. This effect is shown in Figure 2a, for three values of $T$; the $T = 1$ curve is the same as the INSQ curve plotted in Figure 1. Figure 2b shows the three corresponding conditional halting probability functions, $\mathsf{H}_{\mathsf{INSQ}}(i)$.

In support of this choice for the background user behavior, note that it satisfies requirements 1–3, above. In terms of requirement 5,

[2] That is, we exercise artistic licence at this point, and trust that the reader will accept that our intention is to be illustrative rather than prescriptive.




(a) Background $\mathsf{W}_{\mathsf{M}}(i)$ (b) Background $\mathsf{H}_{\mathsf{M}}(i)$ (c) Adjusted $\mathsf{W}'_{\mathsf{M}}(i)$, $T=5$

**Figure 2:** Adding parameters to the INSQ metric: (a) the background weight function $\mathsf{W}_{\mathsf{INSQ}}(T,i)$ for three values of $T$; (b) the corresponding conditional halting probabilities $\mathsf{H}_{\mathsf{INSQ}}(T,i)$; and (c) an example showing the background and adjusted weights when $T=5$ and $r_i = \langle 1,3,4,6,8,12,14,34,37,43,64,82,86,95\rangle$. All scales are logarithmic.

and assuming cascading evaluation through the ranking, the expected search length for a weight distribution $\mathsf{W}_{\mathsf{M}}(i)$ is given by

$$E = \sum_{i=1}^{\infty} i \cdot \mathsf{L}_{\mathsf{M}}(i) = \sum_{i=1}^{\infty} i \cdot \frac{\mathsf{W}_{\mathsf{M}}(i) - \mathsf{W}_{\mathsf{M}}(i+1)}{\mathsf{W}_{\mathsf{M}}(1)} = \frac{1}{\mathsf{W}_{\mathsf{M}}(1)}\,.$$

Equation 2 then means that for the parameterized INSQ metric

$$E = 4T^2 \left( \frac{\pi^2}{6} - \sum_{i=1}^{2T-1} \frac{1}{i^2} \right) \approx 2T + 0.5\,.$$

That is, a user seeking $T$ answers is modeled as looking, on average, at around $2T+0.5$ documents before concluding their search. For example, the expected numbers of documents examined by $T=1$, $T=5$, and $T=25$ searches (as per the weights plotted in Figure 2a) are 2.58, 10.52, and 50.50 respectively. We believe that this relationship between $T$ and expected search length for the adaptive variant of INSQ helps get us to first base (or even beyond) in terms of "intuitive plausibility".

The default "no relevant documents encountered" behavior embodied in $\mathsf{C}_{\mathsf{INSQ}}(T,i)$ is then modified (requirement 4) by a discounting factor that, as the user gets closer to their goal of finding $T$ relevant documents, increases the probability of the search terminating at any particular depth. One way this can be done is to note that once the $i$th document has been inspected, the user is now anticipating finding $T - \mathsf{Rel}(i)$ relevant documents where, as before, $\mathsf{Rel}(i) = \sum_{j=1}^{i} r_i$ is the total relevance achieved through to depth $i$. A possible formulation for an adjusted conditional continuation probability is to then use

$$T_i = \max\{0, T - \mathsf{Rel}(i)\}\,,$$

as an estimate of the volume of relevance still anticipated, and take

$$\mathsf{C}'_{\mathsf{INSQ}}(T_i,i) = \frac{(i+2T_i-1)^2}{(i+2T_i)^2}\,. \tag{3}$$

Figure 2c shows the effect of these changes on an example ranking processed when $T=5$ answers are anticipated. The particular ranking used is rich in relevant documents near the top, and so, compared to the model established by the background probabilities, the user is more likely to halt early. That propensity translates into an adaptive weighting function $\mathsf{W}'_{\mathsf{INSQ}}(T,i)$ that can only be computed once the ranking is given. In the case of the example, the computed effectiveness score rises from 0.350 to 0.502 as a result of the adaptation.

| | insq5 | insq10 | rr | insq | p10 | rbp95 | sdcg100 | ap |
|---|---|---|---|---|---|---|---|---|
| insq5 | **67.5** | 65.9 | 52.4 | 61.9 | 63.3 | 64.9 | 64.2 | 62.1 |
| insq10 | — | **69.0** | 51.8 | 60.8 | 63.9 | 67.6 | 66.6 | 64.2 |

(a) Percentage in SSA category

| | insq5 | insq10 | rr | insq | p10 | rbp95 | sdcg100 | ap |
|---|---|---|---|---|---|---|---|---|
| insq5 | — | 96.6 | 85.1 | 94.3 | 96.0 | 95.0 | 92.7 | 88.7 |
| insq10 | — | — | 83.1 | 91.6 | 95.7 | 97.9 | 95.3 | 90.9 |

(b) Class agreement for SSA category

**Table 4:** TREC-8 data: (a) percentages of system pair comparisons in the SSA categories for selected metric combinations; and (b) class agreements. These values correspond to the numbers on and above the diagonals in Tables 1a and 1b respectively.

| | insq5 | insq10 | rr | insq | p10 | rbp95 | sdcg100 | ap |
|---|---|---|---|---|---|---|---|---|
| insq5 | **60.8** | 58.0 | 44.5 | 52.6 | 54.9 | 55.9 | 52.0 | 50.9 |
| insq10 | — | **61.6** | 42.7 | 50.7 | 54.5 | 58.6 | 54.4 | 53.1 |

(a) Percentage in SSA category

| | insq5 | insq10 | rr | insq | p10 | rbp95 | sdcg100 | ap |
|---|---|---|---|---|---|---|---|---|
| insq5 | — | 94.8 | 80.4 | 90.5 | 91.8 | 90.6 | 86.1 | 82.4 |
| insq10 | — | — | 76.6 | 86.7 | 90.5 | 94.4 | 89.5 | 85.5 |

(b) Class agreement for SSA category

**Table 5:** TREC-10 data, other details as for Table 4, and can be compared with the values on and above the diagonals in Table 2.

Tables 4 and 5 extend Tables 1 and 2 respectively, concentrating on the "above the diagonal" values. Two versions of the adaptive INSQ metric defined by Equation 3 are included, with parameter values $T=5$ and $T=10$, denoted by "insq5" and "insq10" respectively. The column headed "insq" is the static INSQ metric defined by Equation 1, already compared to other static and adaptive metrics in Tables 1 and 2. As can be seen from the corresponding part (a) segments, in terms of their ability to determine significance, INSQ5 behaves somewhat like the shallow metrics RR and P@10, and INSQ10 is somewhat like the deeper ones. This relationship is as expected. In the corresponding part (b) in each table it is notable that the adaptive INSQ-based metrics have higher class agreements with both deep metrics (AP) and shallow ones than do any of the other metrics considered. This is a very encouraging outcome.

## 6. RELATED WORK

There has been a great deal of thought given to effectiveness evaluation over the last decade. Järvelin and Kekäläinen [8] in-

troduced the idea of inner-product top-weighted measures and described both DCG and a normalized variant of it called NDCG that we have not considered here; Moffat and Zobel [10] followed up by describing the RBP metric and formalizing the corresponding user model. Zhang et al. [16] considered a range of static weighted-precision metrics, and showed that RBP with $p = 0.73$ was a good fit with the click densities observed in a commercial search engine click log; Carterette et al. [4] also examine the choice of $p$ in RBP, reiterating that it might vary across both users and queries. Robertson [11] provided a user model for AP; Thomas et al. [13] examine the numeric stability of static metrics when applied to perturbed or degraded rankings; they also note that page boundaries can also be handled by altering the continuation probabilities at appropriate intervals. Zhang et al. [16] also consider page boundaries.

Chapelle et al. [5] examine weighted-precision effectiveness metrics, and argue that the history of what the user experiences as they process the answer list affects the way they address the remainder of the list, and discuss ways in which these adaptive cascade models can be structured; we include that critical requirement in the approach described in this paper. Yilmaz et al. [15] also explore metrics in which the probability of continuing the inspection of documents is conditional on the relevance level of the last document inspected.

Carterette [3] analyzes and categorizes a range of effectiveness metrics, grouping them into four classes; and considers the relationships between weights, halting probabilities, and last viewed probabilities that we have also employed in this work. He then explores the implications of the classification using a range of click and TREC data, concluding that DCG has a range of merits.

Most recently Smucker and Clarke [12] have measured the time taken by users to inspect documents, and argued that a more precise unit of "investment" against which utility is assessed should be search time, rather than documents examined. In a user study of search behavior, Smucker and Clarke [12] demonstrate that short documents require less inspection time than do long ones, and that repeated documents can be evaluated very quickly. Based on these, and other factors, they propose *time-biased gain* as an effectiveness metric, and argue that it better reflects user search behavior.

In other user-focused work, Al-Maskari et al. [1] question the usefulness of deep evaluation metrics, and find that shallow metrics such as P@10 provide better correlation with the experience reported by users. Doubts have also been expressed about the usefulness of AP as a metric by Turpin and Scholer [14], who measured user task completion using degraded rankings; Huffman and Hochster [7] go further, and compare user satisfaction with a simple depth-three effectiveness metric, and find a strong correlation between them.

There has also been investigation into how best to address the complication of diversity, the fact that a query may have multiple interpretations. We do not consider that literature here; the reader is referred to Kanoulas et al. [6] and Ashkan and Clarke [2].

# 7. NEXT STEPS

Our key claim is that effectiveness metrics mirror user models, and hence for the scores assigned by a metric to be convincing, the user model must be a plausible one – in particular, that the "cascade" approach to evaluating a ranking must be informed by the user's intention in issuing the query. With that in mind, we have recently commenced a user study to measure search behavior on a variety of task types, ranging from pure navigational to rich information-seeking ones. We have constructed an instrumented browser, and will monitor explicit user action in terms of queries, click-throughs, and document assessments, in the style also described by Smucker and Clarke [12]; and will be correlating those actions against gaze-tracking behavior captured for each user. In each task users will be presented with answer listings generated via the API of a de-identified commercial search service, with half of the result pages "diluted" by the insertion of attractive but not-relevant documents.

Subjects will also be shown a set of similarly-categorized information needs, and asked (without performing any searching) to estimate the number of documents they think they would need to locate in order to satisfy those information needs.

We believe that this experimental structure will allow testing of our key hypothesis, namely, that as relevant documents are identified, users become more inclined to end their perusal of the answer list, but do so more slowly if they initially sought a high number of answers. We expect to have results early in 2013, including extending the notion of "anticipated relevance remaining" through to multi-query sessions [9], as is hinted at by Table 3.


## *Acknowledgment*

This work was supported by the Australian Research Council.


# 8. REFERENCES


[1] A. Al-Maskari, M. Sanderson, and P. Clough. The relationship between IR effectiveness measures and user satisfaction. In *Proc. SIGIR*, pages 773–774, Amsterdam, July 2007.

[2] A. Ashkan and C. L. A. Clarke. On the informativeness of cascade and intent-aware effectiveness measures. In *Proc. WWW*, pages 407–416, Hyderabad, India, Apr. 2011.

[3] B. Carterette. System effectiveness, user models, and user utility: A conceptual framework for investigation. In *Proc. SIGIR*, pages 903–912, Beijing, China, 2011.

[4] B. Carterette, E. Kanoulas, , and E. Yilmaz. Simulating simple user behavior for system effectiveness evaluation. In *Proc. CIKM*, pages 611–620, Glasgow, Scotland, 2011.

[5] O. Chapelle, D. Metzler, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Proc. CIKM*, pages 621–630, Hong Kong, China, 2009.

[6] C. L. A. Clarke, N. Craswell, I. Soboroff, and A. Ashkan. A comparative analysis of cascade measures for novelty and diversity. In *Proc. WSDM 2011*, pages 75–84, Hong Kong, China, 2011.

[7] S. B. Huffman and M. Hochster. How well does result relevance predict session satisfaction? In *Proc. SIGIR*, pages 567–574, Amsterdam, July 2007.

[8] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Information Systems*, 20(4):422–446, 2002.

[9] E. Kanoulas, B. Carterette, P. D. Clough, and M. Sanderson. Evaluating multi-query sessions. In *Proc. SIGIR*, pages 1053–1062, Beijing, China, July 2011.

[10] A. Moffat and J. Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM Trans. Information Systems*, 27(1):2:1–2:27, Dec. 2008.

[11] S. Robertson. A new interpretation of average precision. In *Proc. SIGIR*, pages 689–690, Singapore, July 2008.

[12] M. D. Smucker and C. L. A. Clarke. Time-based calibration of effectiveness measures. In *Proc. SIGIR*, pages 95–104, Portland, Oregon, Aug. 2012.

[13] P. Thomas, T. Jones, and D. Hawking. What deliberately degrading search quality tells us about discount functions. In *Proc. SIGIR*, pages 1107–1108, Beijing, July 2011.

[14] A. Turpin and F. Scholer. User performance versus precision measures for simple search tasks. In *Proc. SIGIR*, pages 11–18, Seattle, Washington, Aug. 2006.

[15] E. Yilmaz, M. Shokouhi, N. Craswell, and S. Robertson. Expected browsing utility for web search evaluation. In *Proc. CIKM*, pages 1561–1564, Toronto, Canada, 2010.

[16] Y. Zhang, L. A. F. Park, and A. Moffat. Click-based evidence for decaying weight distributions in search effectiveness metrics. *Information Retrieval*, 13(1), Feb. 2010.

# Sentence Length Bias in TREC Novelty Track Judgements

Lorena Leal Bando & Falk Scholer
School of Computer Science
and Information Technology
RMIT University
Melbourne, Australia
{lorena.lealbando,falk.scholer}@rmit.edu.au

Andrew Turpin
Dept. of Computer Science
and Software Engineering
University of Melbourne
Melbourne, Australia
aturpin@unimelb.edu.au

## ABSTRACT

The Cranfield methodology for comparing document ranking systems has also been applied recently to comparing sentence ranking methods, which are used as pre-processors for summary generation methods. In particular, the TREC Novelty track data has been used to assess whether one sentence ranking system is better than another. This paper demonstrates that there is a strong bias in the Novelty track data for relevant sentences to also be longer sentences. Thus, systems that simply choose the longest sentences will often appear to perform better in terms of identifying "relevant" sentences than systems that use other methods. We demonstrate, by example, how this can lead to misleading conclusions about the comparative effectiveness of sentence ranking systems. We then demonstrate that if the Novelty track data is split into subcollections based on sentence length, comparing systems on each of the subcollections leads to conclusions that avoid the bias.

## Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search process

## General Terms

Experimentation

## Keywords

Sentence ranking, query-biased summaries, sentence length

## 1. INTRODUCTION

Automatic text summarisation is a well-studied field that includes a wide range of summary types, and methods to create summaries [11, 19, 21]. One method, extractive summarisation, constructs summaries by excerpting passages of documents that are deemed to be important, and then combines those passages to form the summary. As sentences typically express a single and complete idea, they are the dominant form of passage used in extractive summarisation. Sentences are particularly suitable as the building block of summaries for applications where the summary must be of a specified length. Thus, a key component of extractive summarisation methods is sentence ranking [7, 8, 17, 25, 30], where sentences are ranked according to some measure of their suitability for forming part of a summary, and the top $m$ sentences are joined to create the summary.

Sentence ranking is a similar problem to document ranking, the cornerstone of the information retrieval (IR) discipline. In document ranking, documents in a collection are scored for their similarity to a posed query; for example, just as Google and Bing rank Web pages against a query. The IR field has a long history of rigorous methods for evaluating the effectiveness of document ranking systems, and so those well-established techniques can be adopted to assess sentence extraction methods that form the foundation of current summarisation methods.

The Cranfield methodology establishes a framework to gauge the effectiveness of IR systems in the context of document retrieval [5]. Such a framework involves: a text collection that is a set of documents to be searched; a set of topics which resemble user requests that could be answered using the documents in the collection; and judgements, given by external assessors, that estimate the "relevance" of a document with respect to each topic. Note that relevance is a complex concept and can vary from person to person, but in order to make experiments based on the Cranfield methodology tractable, relevance is typically simplified into a binary scale as determined by one person, and judgements are based only on the topical content or "aboutness" of a document. When comparing two systems, each topic is run by each system, a ranked list of documents is returned by each system, and then the relevance of documents in each list is scored and combined to give a quantitative measure of each system's effectiveness.

TREC 2002 introduced the Novelty track [10], which mainly aimed to investigate approaches to detect non-redundant information at the sentence level. The track supplied separate judgements on individual sentences for both "relevance" to the topics and "novelty". Therefore, these sentence level judgements can be used to evaluate sentence ranking systems that aim to rank based on relevance or novelty.

When assessing sentence ranking systems that are a preprocessor for a summarisation system, one could assume that the sentences should be sorted by "relevance" to a particular topic, and thus the Novelty track judgements can be


*ADCS '12*, December 5-6, 2012, Dunedin, New Zealand.

used directly to score systems. However, depending on how the sentences are used to form a summary, relevance as determined by a single judge at a topic level may not be a valid basis for judging sentence ranking systems. In a Web context, where very short summaries called "snippets" are generated, sentences selected should indicate to a user whether to click through to the underlying document or not. Therefore, "indicativeness" may be a more suitable property on which to base judgements to compare sentence ranking systems. However, previous studies have used the Novelty track data to investigate passage retrieval [16] and summarisation approaches [18]; we therefore also assume that a sentence that is labelled as relevant for an assessor of the Novelty track is also indicative for assembling a summary. Thus, Novelty track relevance assessments are used for evaluating sentence ranking methods that form a pre-processor to snippet generation.

This paper exposes a weakness in using the Novelty track data in this way: namely there is a strong connection between sentence length and relevance. The next section examines the Novelty track data in detail, demonstrating this connection. Section 3 then shows an example of how using the typical Cranfield methodology and the Novelty track data demonstrates that the Vector Space Model adapted for sentences outperforms query-biased sentence selection. If a length component is introduced to each system; however, then the result is reversed and the query-biased system comes out on top. Note that both of these approaches are highly regarded in the literature, and not simply straw men. A second experiment in this section examines the effect of query expansion on the two systems, and the interplay of the length bias in assessing the superior system. Section 4 explains how the Novelty track data could be used to avoid making misleading system comparisons such as those in Section 3. In Section 5 we discuss our results, with conclusions and future work presented in Section 6.

## 2. NOVELTY TRACK DATA

The TREC Novelty track ran from 2002 to 2004 [10, 23, 24], and aimed to study passage retrieval to identify non-redundant content. In the first year of the track, assessors had specific constrains that led the identification of a very small proportion of relevant sentences. These restrictions included: to not select contiguous sentences; to not create topics; and to make judgements towards a short topic description. For these reasons organisers of the track suggested that the outcomes of the Novelty track 2002 should only be regarded as a pilot experiment [10]. Hence, we only use the data from 2003 and 2004 in our experiments.

The 2003 and 2004 Novelty tracks are more homogeneous in terms of constructing topics and gathering judgements [23, 24]. NIST assessors created 50 topics from the AQUAINT newswire collection for each year of the track, and identified relevant documents for the topic by employing the WebPRISE information retrieval system. Novelty 2003 was comprised of 25 relevant documents per topic, while Novelty 2004 tallied more than 25 documents for some topics, as irrelevant documents were intentionally included. Irrelevant documents were added in order to increase the complexity of the task for participants, relative to 2003.

Documents in the AQUAINT collection contain an `id` corresponding to the date of authorship. Relevant documents were sorted according to this identifier, and split into sentences. All document sentences were pooled into a single document for judgement, so an assessor inspected documents chronologically instead of their ranked position provided by the retrieval system. Assessors were asked to distinguish relevant sentences in a topic, and to identify those that were novel. That is, relevance and novelty judgements were made separately. Assessors were able to select any number of relevant and novel sentences per topic or document.

Despite the fact that the main goal of the Novelty track was to study techniques to avoid redundant content, the availability of relevance judgements at the sentence level makes the track appealing for other types of applications. For instance, previous research has employed the Novelty track data to evaluate machine learning approaches for snippet generation [18] and passage retrieval tasks assisted by statistical query expansion [16]. The former work found that the selection of features among Novelty data sets from 2002 to 2004 were not robust. Losada [16], on the other hand, found that query expansion effectively assisted the identification of relevant sentences. However, his findings are applicable for sentences within a set of documents regarding the same topic, rather than individual documents.

### 2.1 Length Bias

Previous research has shown that the relevance of a document tends to increase with its length, since long documents may include information related to not only one but several requests. Consequently, users are prone to select these documents as relevant. The document length bias was investigated in early TREC conferences to determine the effectiveness of a retrieval system [22]. It was found that if a system took into account the length of a document in the ranking process, it could outperform those that did not. This length bias has not previously been investigated for sentence ranking tasks.

Approaches to extractive summarisation rely on the occurrence of significant terms, cue words, query words, or title words of documents for ranking sentences (discussed in more detail in Section 3.1). Therefore, long sentences are more likely to contain these terms in comparison to short sentences, and thus to be scored more highly with respect to being included in summaries.

To avoid ranking long sentences ahead of short sentences purely because of length, sentence scores can be normalised by dividing by the total number of words in each sentence [26]. Normalisation emerged as a mechanism to minimise the effect of retrieving long documents [22] for the document ranking problem. Another approach is to ignore brief sentences, since they might not include relevant content due to their length [13].

### 2.2 Length Bias in the Novelty Track

A simple approach to detect a length bias in the data set is to count the number of words in relevant and irrelevant sentences. It should be noted that assessors did not provide relevance judgements for sentences in irrelevant documents, which were intentionally included in the Novelty 2004. Thus, there is a potential pool of relevant sentences that were not judged as such in the collection. To avoid the confounds that these might introduce in our analysis, we discarded all documents (hence sentences within those documents) that do not contain at least one relevant sentence. This ensures that every sentence in our analysis of the Novelty track 2004

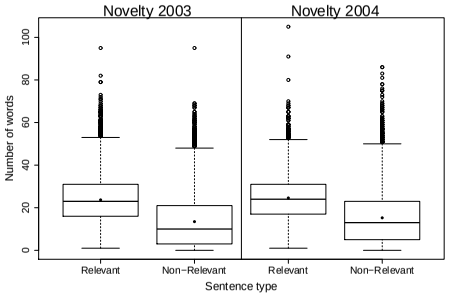



**Figure 1: Distribution of the number of words in each sentence (including stopwords) in the Novelty track data sets (2003 and 2004). Boxes are 25th and 75th percentiles, bar is median, whiskers and circles show extremities, and the dot represents the mean.**

**Table 1: Composition of the two Novelty track collections from 2003 and 2004 for known judged documents. Sentence length is the mean number of words (including stopwords.)**

| | Novelty track | |
|---|---|---|
| Feature | 2003 | 2004 |
| Number of relevant sentences | 15,557 | 8,343 |
| Number of non-relevant sentences | 24,263 | 28,628 |
| Length of relevant sentences | 23.69 | 24.59 |
| Length of non-relevant sentences | 13.47 | 15.26 |
| Number of topics | 50 | 50 |
| Number of documents | 1,250 | 1,070 |

data has been judged.

Figure 1 shows the sentence lengths on the two collections. The mean length of non-relevant sentences is significantly less than the mean length of relevant sentences (*t-test* $p < 0.001$), with details given in Table 1. There is a clear connection between length and relevance in the 2003 and 2004 Novelty track data. The next section examines how this length bias manifests in sentence ranking methods.

## 3. SYSTEM COMPARISONS

This section describes two systems for ranking sentences, and compares them using the Cranfield methodology on the Novelty track 2003 and 2004 data ignoring possible sentence length bias. We then introduce a length bias component into each system and study the effects.

### 3.1 Extractive and Query-biased Summarisation

An extractive summarisation method ranks sentences based on evidence collected either directly from documents or from shallow sentence attributes. The former mechanism involves word frequency statistics to determine significant words from the document [17], to identify title and heading words or cue words [7]. Sentences are scored according to the occurrence of such terms. Early summarisation approaches have employed evidence from documents. However, modern text collections may have metadata information [12] and anchor text [2] available, which can be used for assisting Web page summarisation.

Shallow sentence attributes, on the other hand, are independent from the document vocabulary and concentrate on superficial features such as the position [4] or length of sentences [13], and word formatting [30]. In order to take advantage of both document content and shallow sentence features, these can be merged using a linear combination where constants tune the value that each approach contributes to the final score of a sentence.

Query-biased summarisation (also called query-dependent, query-specific or query-relevant) is a type of extractive summarisation that favours the selection of sentences, or passages, that contain query terms. In large text collections, these summaries are helpful for guiding users to dismiss irrelevant documents and to inspect those that appear likely to be relevant given the summary [25]. Query-biased summarisation can rely on simple heuristics that count for query terms occurrence, or document ranking functions adapted for sentence selection. The following subsections explain both approaches in detail.

#### *Query Term Occurrence*

Tombros and Sanderson [25] introduced a score that depended on the appearance of query terms for the ranking of candidate sentences for summaries of Web search results. This score was computed in a similar fashion as the clusters of significant words proposed by Luhn [17]. The query-biased score for a sentence $s$ is calculated as:

$$\text{QB}_s = \frac{(|qt|)^2}{|q|} \quad (1)$$

where $|qt|$ is the number of unique query terms in sentence $s$ and $|q|$ is total number of words in the query. Other variants for employing query terms occurrence include the count of repeated query terms, and the longest contiguous sequence of query terms in a sentence [27]. In commercial applications, query counting occurrence is used for extracting "query-relevant" parts from documents. These excerpted parts are not only useful for displaying snippets of search engine results, but also for detecting duplicate documents without the need of analysing the whole document [9].

#### *VSM*

A variety of retrieval models have been proposed in the literature such as the Vector Space Model (VSM), the Okapi BM25 similarity function, and Language Models [6]. By treating each sentence as a "document" is straightforward to apply these to score sentences relative to a query [1, 8, 16, 28]. For example, the cosine similarity function in the VSM calculates the Euclidean distance between weighted document and query vectors. That is, the shorter the distance between both vectors, the more similar a query is to a document (or to a sentence). Allan et al. [1] adapted the VSM for determining the similarity of a sentence to a query as follows:

$$R(s|q) = \sum_{t \in q} \log(f_{(t,q)} + 1) \log(f_{(t,s)} + 1) \log\left(\frac{n+1}{0.5 + f_t}\right) \quad (2)$$

where $f_{(t,q)}$ and $f_{(t,s)}$ are the occurrence of term $t$ in query $q$ and sentence $s$, respectively. The number of sentences in the

collection is given by $n$, and $f_t$ is the number of sentences in which the term $t$ appears.

Similar applications of other retrieval models for sentence selection have been studied. Nevertheless, comparisons in previous work have not shown significant differences between the effectiveness of VSM compared to the Okapi BM25 similarity function [16], or Language Models with the Kullback-Leibler divergence [1, 14, 16].

The generation of query-biased summaries is not restricted to statistical methods; machine learning approaches can also be used [18, 29]. However, the focus of this paper is not an exhaustive comparison of query-biased summarisation methods, so we restrict our attention to two widely-used approaches: the query-biased score (QB) and the VSM model, described above.

## 3.2 Baseline Systems

The QB approach consists of counting occurrences of query terms in a sentence as defined in Equation 1. The VSM approach employs the vector space model adaptation for sentence retrieval as applied by Allan et al. [1]. Since terms are generally not repeated in typical Web search queries, the VSM is simplified as follows:

$$R(s|q) = \sum_{t \in q} \log(f_{(t,s)} + 1) \log\left(\frac{n+1}{0.5 + f_t}\right) \qquad (3)$$

As we are addressing a single-document summarisation problem, the parameter $n$ in this equation is the number of sentences in a given document, instead of the number of sentences in a collection as used by Allan et al. [1].

Both the QB and VSM approaches allow the scoring of all sentences in a document, and the top $m$ sentences are selected for inclusion in a summary. Ties in scores can be broken by choosing sentences that occur closer to the beginning of the document. These two baseline ranking approaches are identified as QB-Pos and VSM-Pos. A second simple way to resolve ties is in the favour of longer sentences. These approaches are labelled QB-Len and VSM-Len.

## 3.3 Results

The Novelty 2003 and 2004 track supplied relevance judgements at the sentence level, thus a simple way to measure the performance of ranking methods is to calculate the proportion of returned sentences that are relevant in a document. Similar to document ranking evaluation approaches, we adopt P@$m$ as the measure to quantify the performance of summarisation methods, where $m$ is the number of returned sentences. In our experiments we use $m$=2, as the aim is to assemble short excerpts. Thus, for any topic we can average the P@2 for each document for that topic, and compare the means between different sentence ranking schema. From the Novelty data set, we included all documents that have at least $m$ relevant and $m$ non-relevant sentences as part of our test collection.

The top two rows of Table 2 show the results of the two baseline methods using the "title" field of the TREC Novelty topics as a query. The title averages a length of three words for both Novelty 2003 and 2004, similar to current Web queries [3]. While VSM-Pos outperformed QB-Pos significantly for Novelty 2003 (t-test, $p < 0.001$), the percentage change for Novelty 2004 is not significant ($p > 0.05$).

Rows four and five show the two baseline methods, but here tied sentence-ranking scores are resolved based on decreasing sentence length. Note that the P@2 values are higher (between 7% and 26%), which is to be expected as now longer sentences are being favoured, and we know that, in general, longer sentences are more likely to be relevant in these collections. All increases are statistically significant (*t-test*, $p < 0.001$). Not only are all four numbers higher than when not using length to break ties, but now the QB based system is better than the VSM system (*t-test*, $p = 0.002$ and $p = 0.006$ for Novelty 2003 and 2004, respectively). That is, taking length into account has reversed the result of the original experiment.

**Table 2: P@2 results of the four methods with original and expanded queries. The method Len ignores the query and consider longer sentences, and a method that randomly selects two sentences (Random). An $**$ indicates statistical significance (paired t-test) of $p < 0.001$ and $*$ of $p < 0.01$.**

| Method | Novelty 2003 | Novelty 2004 |
|---|---|---|
| Original query *(title)* | | |
| QB-Pos | 0.61 | 0.52 |
| VSM-Pos | 0.68 | 0.53 |
| Change | 10%** | 2% |
| QB-Len | 0.77 | 0.58 |
| VSM-Len | 0.75 | 0.56 |
| Change | -3%* | -3%* |
| Only length | | |
| Len | 0.72 | 0.52 |
| Random | 0.44 | 0.27 |
| Expanded query *(title and narrative)* | | |
| QB-Pos | 0.73 | 0.60 |
| VSM-Pos | 0.75 | 0.62 |
| Change | 1% | 3% |
| QB-Len | 0.79 | 0.63 |
| VSM-Len | 0.75 | 0.62 |
| Change | -4%** | -1% |

As a point of comparison, we ranked sentences using only their length and ignore any score that involves the query. This simple baseline is called Len, row seven in the above table. It can be noted that for the Novelty 2003 VSM-Pos, which outperformed QB-Pos, performed more poorly than Len. This confirms a strong length bias in the relevance assessments in this data set as suggested by Metzler and Kanungo [18] when using machine learning approaches on the Novelty 2003 data. In the case of the Novelty 2004, the length effect is more moderate compared to QB-Pos and VSM-Pos.

### *Query Expansion*

It has been suggested that automatically expanding a query can assist in the selection of sentences for passage retrieval tasks [16]. We were also investigating the use of query expansion to improve sentence ranking for snippet generation

when we came across the length bias in the TREC Novelty track 2003 and 2004 data described in this paper. While there are some sophisticated query expansion methods we could apply to QB-POS and VSM-POS, in this paper we add the "narrative" field of the TREC Novelty topics to the "title" field as ready-made expanded queries. The narrative field of a TREC Novelty topic consists of a verbose statement of the information need that corresponds to the short title query; using this as a proxy for query expansion is therefore equivalent to considering a case where a relevance feedback system has elicited an extended description of an information need from a user.

The bottom section of Table 2 shows the results for these queries. In all cases, the systems using the expanded queries scored higher than their non-expanded counterparts. And again, ignoring length puts VSM ahead of QB as the method of choice, while adding length reverses the result. Row eight in the same table shows the performance of randomly selecting sentences (RANDOM). This method significantly achieved poorly in comparison to the LEN approach and both VSM and QB using original or expanded queries ($p < 0.001$).

## 4. ISOLATING SENTENCE LENGTH BIAS

While it has been proposed that short sentences could simply be ignored in the construction of summaries [13], for query-biased summaries in space-limited environments such as search result pages, short sentences can be valuable. We therefore investigate how to isolate the sentence length bias factor for evaluation purposes. Our approach is similar to that used by Singhal et al. [22] for document retrieval. In their approach, documents were grouped by their length (measured in bytes). In contrast, we bucketed sentences of specific lengths measured in words to attempt to counter the effects of a length predisposition.

We obtained the average length ($\mu$) of relevant and irrelevant sentences in the Novelty data, as well as the standard deviation ($\sigma$). This is $\mu = 17$ and $\sigma = 12$ words for both Novelty track 2003 and 2004 (according to sentence statistics listed in Table 1). Based on this information we classified sentences in three buckets: $l_1$, where sentences have between 5 ($\mu - \sigma$) and 13 words; $l_2$ contains sentences from 14 to 20 words; and $l_3$, containing sentences between 21 to 29 ($\mu + \sigma$) words. Given that each bucket contains sentences of different lengths, the amount of information for each sentence may vary from bucket to bucket. A summary composed of two sentences from bucket $l_3$ is longer – and potentially more indicative – than a summary composed of two sentences from bucket $l_1$. To account for this, we adapt the value of $m$ in P@$m$ for each bucket. Specifically, we used P@4, P@3 and P@2 for measuring effectiveness in each of three buckets $l_1$, $l_2$ and $l_3$, respectively.

Table 3 lists the number of documents that exist in each bucket. This can be seen as splitting the Novelty data set into several subcollections, where a document has at least $m$ relevant sentences of length $l_i$. For comparison purposes, for each bucket we collected the $m$ longest sentences (LEN approach), and randomly selected $m$ sentences of length $l_i$ (RANDOM approach).

The results of evaluating the different summarisation approaches using the length-bucketing approach are shown in Table 4. Contrary to the results in Section 3.3, we did not find significant differences between QB-POS and VSM-POS when employing either the original query (rows 1 and 2), or its expanded version (rows 4 and 5). By analysing buckets of length $l_2$ and $l_3$ in the Novelty track 2003 data, the expansion did not reveal any improvement over the original query when the sentence length feature is isolated. However, for the same buckets in the Novelty track 2004 the differences were significant (t-test, $p < 0.05$). For short sentences (bucket $l_1$), the difference when using query expansion is significant ($p < 0.05$) for both the Novelty 2003 and 2004 data, with percentage changes from 4% to 8%.

**Table 3: Number of documents in each bucket to compute P@$m$.**

| Bucket | $m$ | Novelty 2003 | Novelty 2004 | Total documents |
|---|---|---|---|---|
| $l_1$ | 4 | 629 | 588 | 1217 |
| $l_2$ | 3 | 722 | 682 | 1404 |
| $l_3$ | 2 | 1020 | 983 | 2003 |

Given that QB-POS is not significantly different from VSM-POS, we use the former approach for comparison against LEN. For buckets of length $l_1$, LEN performs significantly better than using the expanded query in both data sets ($p < 0.05$). For the remaining buckets, we noted that QB-POS with query expansion performs significantly better than the LEN method ($p < 0.001$). Finally, we observed that the LEN method against the RANDOM differs performance using buckets of length $l_1$ in the Novelty track 2003 and 2004 ($p < 0.001$). For the other buckets both methods achieved similar P@$m$ scores ($p > 0.05$).

## 5. DISCUSSION

We have demonstrated that there is a length bias in the TREC Novelty track data, where relevant sentences tend to be longer sentences. As a demonstration of how this bias could lead to misleading claims of system effectiveness, we report novel system comparisons on two leading methods for the generation of query-biased summarisation. The thesis of this paper is not to promote one or other of the methods studied, but rather to indicate a potential weakness in using the Novelty track sentence relevance judgements to assess systems that rank sentences.

It can be observed that, in general, a RANDOM approach did not outperform the QB- or VSM-based methods. The exceptional case when this occurs is in the Novelty 2003 data for buckets of length $l_1$. This suggests that for short sentences (5-13 words) in this data set, other constrains should be applied to more accurately discern relevant sentences. The values to define thresholds in buckets were gathered from sentences in the Novelty track as outlined in Section 2.2. Given that a summary is restricted in size, we assumed that 4, 3 and 2 sentences were representative for their corresponding buckets. However, we did not explore other parameter settings for the optimal number of words to be considered in each bucket.

In this paper we have focussed solely on evaluating the sentence ranking problem using TREC Novelty track data. The Text Analysis Conference (TAC), formerly known as the Document Understanding Conference (DUC), has investigated different summarisation styles and proposed several intrinsic evaluation methodologies since 2001. Such methodologies assessed automatic summaries in terms of vocabulary

Table 4: P@$m$ values for buckets of sentences of length $l$.

| | | Novelty 2003 | | | Novelty 2004 | | |
|---|---|---|---|---|---|---|---|
| | | P@2 | P@3 | P@4 | P@2 | P@3 | P@4 |
| | | $l_3$ | $l_2$ | $l_1$ | $l_3$ | $l_2$ | $l_1$ |
| Original (*title*) | QB-Pos | 0.709 | 0.613 | 0.306 | 0.489 | 0.383 | 0.188 |
| | VSM-Pos | 0.705 | 0.614 | 0.308 | 0.492 | 0.384 | 0.187 |
| | Change | -0.6% | 0.2% | 0.6% | 0.5% | 0.2% | -1.0% |
| Expanded (*title and narrative*) | QB-Pos | 0.708 | 0.616 | 0.327 | 0.520 | 0.401 | 0.196 |
| | VSM-Pos | 0.699 | 0.613 | 0.333 | 0.512 | 0.404 | 0.198 |
| | Change | -1.3% | -0.4% | 1.7% | -1.4% | 0.6% | 1.5% |
| Len | — | 0.643 | 0.572 | 0.398 | 0.412 | 0.338 | 0.209 |
| Random | — | 0.636 | 0.562 | 0.335 | 0.403 | 0.327 | 0.175 |

overlap [15], or content matching units [20], against a set of ideal summaries. This set is comprised of abstracts authored by assessors who may merge ideas into a single sentence or paraphrase content, for example. Hence, the framework provided by TAC/DUC cannot be used straightforwardly to evaluate sentence ranking methods. Despite the fact that the Novelty track offers relevance judgements of sentences, further research is required to evaluation on such assessments reliable, particularly for applications where the use of these judgements deviate from *ad-hoc* and novelty tasks, which were the main aims of the track.

## 6. CONCLUSIONS AND FUTURE WORK

In this paper we investigated the way in which the length of a sentence affects the selection of relevant sentences, specifically in the Novelty track data, which has been used in the IR field to evaluate sentence ranking and snippet generation. This fact calls into question past conclusions on the effectiveness of sentence ranking approaches for tasks such as summarisation or passage retrieval that were based on this data set.

We found that using a short baseline query, similar to current Web queries, ranking methods performed significantly better when employing the sentence length component. Simulating a simple query expansion approach by using the narrative field of Novelty track topics showed that significant improvements from 7% to 26% can be achieved when ignoring the impact of sentence length. However, this advantage disappears when sentence length is included as a component in the ranking method.

We proposed an alternative method for evaluating sentence ranking methods for the construction of query-biased summaries by measuring P@$m$ of sentences of similar length. This approach avoids any length predisposition of sentences. Under this controlled evaluation framework, both original and expanded query-biased approaches were shown to outperform a baseline where sentences were selected only based on length. Similarly, both original and expanded queries were in general able to outperform a random selection of sentences, except for the 2003 Novelty data for short sentences (bucket $l_1$). We plan to study other formal approaches of query expansion in future work.

Furthermore, our analysis demonstrated that relevance judgements from the Novelty track might not be entirely suitable for evaluating summarisation approaches. These assessments were designed for *ad-hoc* tasks and it is not clear whether they can straightforwardly be treated as surrogates of indicative content. For example, snippets that are displayed by search engines need to be concise, so that users can make a decision to click on a result or to keep reading the list of results. In a follow-up study, we plan to directly investigate the indicative value of relevant sentences in the Novelty track for the assembling of query-biased summaries, and the relation between sentence length and indicative content.

## 7. ACKNOWLEDGEMENTS

This research is supported in part by ARC Grant FT0991326 (AT) and CONACyT scholarship (201099).

## 8. REFERENCES

[1] J. Allan, C. Wade, and A. Bolivar. Retrieval and Novelty Detection at the Sentence Level. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, 314-321, ACM Press, 2003.

[2] E. Amitay and C. Paris. Automatically Summarising Web Sites: is there a way around it? In *Proceedings of the ninth international conference on Information and knowledge management*, 173-179, ACM Press 2000.

[3] M. Bendersky and W. Bruce Croft. Analysis of Long Queries in a Large Scale Search Log. In *Proceedings of the 2009 workshop on Web Search Click Data*, WSCD '09, 8-14, ACM Press, 2009.

[4] R. Brandow, K. Mitze, and L. F. Rau. Automatic Condensation of Electronic Publications by Sentence Selection. *Information Processing & Management*, 31(5):675-685, 1995.

[5] C. Cleverdon. The Cranfield Tests on Index Language Devices. *ASLIB Proceedings*, 19:173-194, 1967.

[6] W. B. Croft, D. Metzler, and T. Strohman. *Search Engines: Information retrieval in practice*. Addison Wesley, 2009.

[7] H. P. Edmundson. New Methods in Automatic Extracting. *Journal of the ACM*, 16(2):264-285, 1969.

[8] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing Text Documents: Sentence Selection and Evaluation Metrics. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 121-128. ACM Press, 1999.

[9] B. Gomes and B. T. Smith. Detecting Query-specific Duplicate Documents, Patent No. 6,615,209 B1, 2003.

[10] D. Harman. Overview of the TREC 2002 Novelty Track. In *Proceedings of TREC 2002*, 2002.

[11] K. Spärck Jones. Automatic Summarising: The state of the art. *Information Processing & Management*, 43(6):1449-1481, 2007.

[12] M. Kaisser, M. A. Hearst, and J. B. Lowe. Improving Search Results Quality by Customizing Summary Lengths. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, 701-709. ACL, 2008.

[13] J. Kupiec, J. Pedersen, and F. Chen. A Trainable Document Summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, 68-73. ACM Press, 1995.

[14] X. Li and W. B. Croft. Novelty Detection Based on Sentence Level Patterns. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, 744-751, ACM Press, 2005.

[15] C. Y. Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the workshop on text summarization*, 74-81, 2004.

[16] D. E. Losada. Statistical Query Expansion for Sentence Retrieval and its Effects on Weak and Strong Queries. *Information Retrieval*, 13(5):485-506, 2010.

[17] H. P. Luhn. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2(2):159-165, 1958.

[18] D. Metzler and T. Kanungo. Machine Learned Sentence Selection Strategies for Query-biased Summarization. In *Proceedings of SIGIR Workshop on Learning to Rank for Information Retrieval*, 40-47, 2008.

[19] A. Nenkova and K. McKeown. Automatic Summarization. *Foundations and Trends in Information Retrieval*, 5(2-3):103-233, 2011.

[20] A. Nenkova and R. Passonneau. Evaluationg Content Selection in Summarization: The Pyramid Method. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 145-152, 2004.

[21] C. D. Paice. Constructing Literature Abstracts by Computer: Techniques and Prospects. *Information Processing & Management*, 26(1):171-186, 1990.

[22] A. Singhal, G. Salton, M. Mitra, and C. Buckley. Document Length Normalization. *Information Processing & Management*, 32:619-633, 1996.

[23] I. Soboroff. Overview of the TREC 2004 Novelty Track. In *The Thirteenth Text Retrieval Conference (TREC 2004)*, 2004.

[24] I. Soboroff and D. Harman. Overview of the TREC 2003 Novelty Track. In *Proceedings of the Twelfth Text Retrieval Conference (TREC 2003)*, 2003.

[25] A. Tombros and M. Sanderson. Advantages of Query biased Summaries in Information Retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 2-10. ACM, 1998.

[26] Y. Tsegay, S. Puglisi, A. Turpin, and J. Zobel. Document Compaction for Efficient Query Biased Snippet Generation. In Mohand Boughanem, Catherine Berrut, Josiane Mothe, and Chantal Soule-Dupuy, editors, *Advances in Information Retrieval*, volume 5478 of *Lecture Notes in Computer Science*, 509-520. Springer Berlin / Heidelberg, 2009.

[27] A. Turpin, Y. Tsegay, D. Hawking, and H. E. Williams. Fast Generation of Result Snippets in Web Search. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 127-134. ACM Press, 2007.

[28] R. Varadarajan and V. Hristidis. A System for Query-specific Document Summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, 622-631, ACM Press, 2006.

[29] C. Wang, F. Jing, L. Zhang, and H. Zhang. Learning Query-biased Web Page Summarization. In *Proceedings of the 16th ACM conference on Conference on information and knowledge management*, 555-562. ACM Press, 2007.

[30] R. W. White, J. M. Jose, and I. Ruthven. A Task-oriented Study on the Influencing Effects of Query-biased Summarisation in Web Searching. *Information Processing & Management*, 39(5):707-733, 2003.

# Multi-Aspect Group Formation using Facility Location Analysis

Mahmood Neshati, Hamid Beigy
Computer Engineering Department
Sharif University of Technology
{neshati, beigy}@ce.sharif.edu

Djoerd Hiemstra
Computer Science Department
University of Twente
d.hiemstra@utwente.nl

## ABSTRACT
In this paper, we propose an optimization framework to retrieve an optimal group of experts to perform a given multi-aspect task/project. Each task needs a diverse set of skills and the group of assigned experts should be able to collectively cover all required aspects of the task. We consider three types of multi-aspect team formation problems and propose a unified framework to solve these problems accurately and efficiently. Our proposed framework is based on Facility Location Analysis (FLA) which is a well known branch of the Operation Research (OR). Our experiments on a real dataset show significant improvement in comparison with the state-of-the art approaches for the team formation problem.

## Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Retrieval models

## General Terms
Algorithms, Experimentation

## Keywords
Multi aspect Team Formation, Expert Matching, Expert Finding, Facility Location Analysis

## 1. INTRODUCTION
Expert group formation has recently attracted a lot of attention in Information Retrieval and Data management communities. Since the assignment of experts to a task/project must be based on both the required skills of the project and knowledge about the expertise of all candidate experts, it is not an easy task and it is challenging to optimize the assignment. In real scenarios, several and sometimes diverse skills are needed to perform a project successfully and completely. Generally, these required skills are implicitly expressed in project descriptions. Besides the required skills of a project, in many cases, the relevant skills of experts are also implicitly reflected in their resume. The main challenges of the expert matching problem are:

1) Textual description of projects can only implicitly express the required skills of them, thus a method is needed to transform the textual description of a project into the set of required skills of that project.
2) Similarly, because of implicit notion of expertise, a method is needed to transform the expertise documents (e.g. resume, professional profile etc.) of each expert into the set of his/her skills.



*ADCS '12*, December 5-6, 2012, Dunedin, New Zealand. Copyright 2012 ACM 978-1-4503-1411-4/12/2012 …$15.00.

3) In an ideal expert group formation, all required skills of a project should be covered by the union of the skills of the assigned group members in a complementary manner (i.e. *Coverage condition*).
4) In an ideal expert group formation, besides covering all required skills of a project, it is preferable that each member of the assigned group individually be able to cover as many as possible the required skills of the project (i.e. *Confidence condition*)
5) Forming multiple dependent expert groups, each expert can only be involved in a limited number of projects. In other words, in formation of multiple expert groups with limited recourses (i.e. experts), the *load balancing* condition should be considered.

While all above conditions are natural and practical, the combination of these conditions in a real application can be challenging. Specifically, with a limited number of available experts, simultaneously maximizing the confidence and coverage of the assigned groups is an interesting and also a non-trivial problem.

As a case study for the expert group formation problem, we consider the problem of review assignment. Review assignment is a common task that many people such as conference organizers, journal editors, and grant administrators would have to do routinely. In this problem, top-*k* relevant reviewers (i.e. a group of experts with *k* members) should be assigned to each paper such that all above mentioned criteria are satisfied. Specifically, 1) the required skills for reviewing a paper can be explicitly determined by some keywords or should be inferred from the abstract/body of the paper. 2) The related research areas/skills of each reviewer (i.e. expert) can be expressed explicitly by some keywords or be inferred from his/here previous papers. 3) Ideally the assigned group of reviewers for each paper should be able to cover all required aspects of that paper. 4) It is preferable that each assigned reviewer of a paper be able to cover all aspects/topics of the paper. 5) In a real conference, each member of program committee (i.e. expert) can only be involved in the review process of a limited number of papers.

In this paper, we formalize the expert matching problem within the unified framework of *Facility Location Analysis (FLA)* taken from Operation Research [1], as a way to account and optimize the expert assignment. We show that our proposed method can improve the performance of expert matching in comparison with the state-of-the-art techniques for multi aspect/skill expert matching such as *Greedy Next Best* [2] and *Integer linear programming* [3].

In our proposed framework, we consider the top-*k* reviewers of each paper as the desirable facilities to be placed as close as possible to their customers (i.e. topics). According to different conditions of the expert matching problem, we define three

problems that all can be solved by the proposed frame work of FLA.

In these problems, given a set of *N* papers and *M* reviewers, each paper should be assigned to a group of exactly *k* members.

1- *Implicit Aspects-Unconstraint Matching (Problem 1):* In this problem, we assume the aspects (i.e. required skills) of each paper are implicitly represented in the abstract of the paper and the skills of each reviewer can be inferred from the expertise document of that specific reviewer. Generally, the expertise document of an expert can be his/her resume but in this paper, we consider the concatenation of one's publications as his/here expertise document. In this problem, each paper should be assigned to a group of $k$ reviewers such that the skill coverage and confidence of the assigned group be maximal. However, in this problem, there is no limitation on the capacity of reviewers (i.e. arbitrary number of papers can be assigned to a reviewer).

2- *Explicit Aspects-Constraint Matching (Problem 2)*: In this problem, we assume that the set of the required skills of a paper and also the set of the relevant skills of a reviewer are explicitly determined (for example by a set of predefined keywords). In this problem, each paper should be assigned to a team of $k$ reviewers such that: 1) in an ideal matching, all aspects of all papers should be covered by the skills of the assigned groups. 2) In an ideal matching, each member of the assigned groups for a paper should be able to cover all required skills of that specific paper and finally 3) each reviewer should get only a limited and predefined number of papers to review.

3- *Implicit Aspects- Constraint matching (Problem 3)*: This problem is the combination of the first and the second problems. In this problem, we assume that the notion of aspects/skills of papers and experts are implicit and on the other hand, each expert has a limited capacity to review the assigned papers. The goal of this problem is to maximize the coverage and the confidence of the assigned groups while the load balancing condition is satisfied.

All above mentioned problems are modeled using the unified framework of facility location analysis. Our experiments demonstrate that this framework outperforms the current state of the art algorithms of expert matching.

## 2. Facility Location Analysis

Facility location analysis is a branch of operations research[1] and computational geometry concerning itself with mathematical modeling and solution of problems concerning optimal placement of facilities in order to minimize transportation costs, avoid placing hazardous materials near housing, outperform competitors' facilities, etc. *Desirable k-facility placement* [4] is a type of facility location problems which concerns with the selection of the *k* optimal locations among *P* candidate locations to build *k* facilities such that the total cost of setup of these facilities and the transportation cost of the customers would be minimal. The goal of optimization in this problem is two-fold:

1. To minimize the total cost of opening those facilities, and,

2. To minimize the weighted distances from the customers locations to their closest facilities.

Various types of the facility location problems are defined in the literature [1] for different usages. Two main types of these problems are *uncapacitated* and *capacitated* facility location placements that can be useful to model the expertise matching problem. In this paper, we formally model the *unconstraint* (i.e. problem 1) and *constraint* multi-aspect/skill expertise matching (problem 2 and 3) using uncapacitated and capacitated facility location placement respectively. In the following subsections, we introduce these problems as well as their approximate and exact solutions.

### 2.1 Uncapacitated Facility Location Analysis (UFLA)

In Uncapacitated Facility Location (UFLA) problem, $k$ facility locations should be selected among *N* available facility locations; such that while each customer is assigned to its nearest facility, the overall cost of building all facilities is minimal. Considering the general definition of facility location problem, an arbitrary number of customers can be assigned to a facility. In other words, there is no constraint on the assignment of the customers to the facilities. The overall cost of building *k* facilities can be defined as follows:

$$cost(S) = \overbrace{\lambda \sum_{i=1}^{k} cost(f_i)}^{Building\ Cost} + \overbrace{(1-\lambda) \sum_{j=1}^{m} demand(c_j) \min_{f \in S} D(f, c_j)}^{Communication\ Cost} \quad (1)$$

In this equation, $S = \{f_1, \dots f_k\}$ indicates the set of selected facilities, $cost(f_i)$ is the opening cost of facility $f_i$, $D(f, c_j)$ indicates the distance between the customer $c_j$ and facility $f$, $demand(c_j)$ indicates the demand of customer $c_j$ and $\lambda$ is a parameter in [0,1]. According to the above objective function, the total cost of opening *k* facilities equals the sum of the *Building Cost* (i.e. the first summation) and the *Communication Cost* (the second summation). Figure 1 illustrates an instance of uncapacitated facility location problem in which the location of the customers and the facilities are indicated by circles and squares respectively. Assuming equal building cost for all candidate locations (i.e. $cost(f_i) = cost(f_j), i, j \in \{1,2, \dots 5\}$), the optimal 3 facility locations (among 5 available candidate locations) and also the assignment of the customers to their nearest location is illustrated in Figure 1. Please note that only 3 facilities can be selected in the optimal solution of Figure 1.

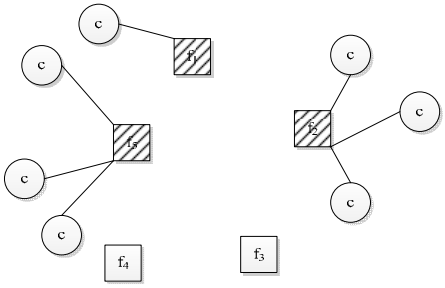



**Figure 1. Uncapacitated Facility location problem- Optimal facilities are indicated by the dashed texture**

The UFLA is in general NP-hard, which can be proved by reduction, for example, from the set cover problem [1]. Since this problem has an explicit objective function, it is possible approximately optimize it using *Greedy Local Search* (GLS), a.k.a. Hill Climbing, as shown in Algorithm 1. The algorithm first initializes set S (i.e. the solution set) with a set of *k* random facilities and then iteratively refines S by swapping a facility location in S and an available non-selected location in D (D indicates the set of candidate facility locations), until the process

converges. Finally, the $k$ facility in S is an approximate solution for the problem.

**Algorithm 1. Greedy Local Search for UFLA problem**

Input: D (set of candidate facility locations),
$k$ (the cardinality of solution set)
Output: S → top-k facility locations

```
S ← {d_1, ... d_k}
repeat
  for d ∈ S do
    for d′ ∈ D\S do
      S′ ← (S\{d}) ∪ {d′}
      if Cost(S′) < Cost(S) then
        S ← S′
      end
    end
  end
until S does not change;
```

## 2.2 Capacitated Facility Location Analysis

Capacitated Facility location analysis (CFLA) is another type of facility location problems that has the same objective function similar to the UFLA. However, in CFLA, each facility has a limited capacity to serve the assigned customers and therefore only a limited (and predefined) number of customers can be assigned to a facility. Figure 2 illustrates the same customer and facility locations indicated in Figure 1. Assuming equal building cost and equal capacity of 2 for each facility, Figure 2 indicates the optimal top 3 facilities for this CFLA. As indicated in this figure, each facility is responsible to serve to *at most* 2 customers and similar to the UFLA problem each customer is assigned to the nearest *opened*/*selected* facility location. Clearly, this problem has no solution for $k$=3 and capacity =2 because the number of customers is bigger than 3*2=6.

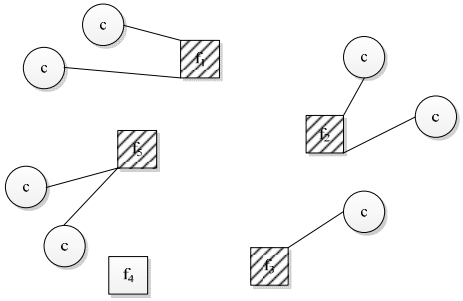



**Figure 2. Capacitated Facility location problem- Optimal facilities are indicated by the dashed texture**

While CFLA problem is in general NP-hard, various approximation algorithms [5][6] are proposed for this problem. Specifically, [6] proposed an efficient linear programming solution for this problem. We can use the approximation algorithms for modeling the expert matching problem, but because of the small size of the problem in our real applications, we chose to exactly solve the matching problem following the idea of linear programming. The explanation of our solution for constraint expert matching based on linear programming is described in section 3.2.

# 3. Multi Aspect Expert Matching

In this section, we describe how to model the expert matching problems using the facility location framework. The list of symbols used in this paper is represented in Table 1.

**Table 1. Notations**

| Symbol | Description |
|---|---|
| $M$ | number of reviewers/experts |
| $N$ | number of papers/projects |
| $T$ | number of aspects/topics |
| $e_i$ | one expert |
| $p_j$ | one paper |
| $k$ | Size of assigned group |
| c | Capacity of each reviewer |

## 3.1 Implicit Aspects- Unconstraint matching

The first problem of expert matching (i.e. *Implicit Aspects-Unconstraint matching)* concerns with the assignment of $N$ papers to $M$ reviewers such that the following conditions are satisfied:

**$C_1$**- Each paper should be assigned to a group of exactly $k$ reviewers.

**$C_2$**- (*Maximal Coverage*) - In an ideal matching, all aspects/topics (i.e. required skills) of each paper $p_j$ should be covered by the assigned group of experts to that paper.

**$C_3$**- (*Maximal* Confidence) - In an ideal matching, each assigned reviewer $e_i$ to paper $p_j$ should have all the required skills of paper $p_j$.

In this problem, the notion of related aspects of papers and reviewers is implicit, making it difficult to maximize the aspect coverage. We assume that the related aspects of a paper can be inferred from its abstract and the skills/aspects of reviewers can be represented by his/her sample publications. We use the concatenation of a reviewer's publications as his/here expertise document. To maximally cover multiple aspects of papers, we try to find a topic representation for each paper and reviewer. Following the idea of reviewer modeling introduced in[2], we can assume that there is a space of $T$ topic aspects, each characterized by a unigram language model such that the papers and the expertise documents can be represented as the mixture of these topics. Let $\tau = (\tau_1, \dots, \tau_k)$ be a vector of topics. $\tau_i$ is a unigram language model and $p(w|\tau_i)$ is the probability of word $w$ according to the topic $\tau_i$. Given $M$ reviewer's expertise documents, we can learn arbitrary number of latent topic/aspects using Probabilistic Latent Semantic Analysis (PLSA) [7]. Let $R = \{r_1, \dots r_n\}$ be the set of expertise documents (i.e. document $r_i$ is the expertise document of reviewer $e_i$), the log likelihood of the expertise document collection according to the PLSA is:

$$\log P(R|\tau) = \sum_{i=1}^{M} \sum_{w \in V} c(w, r_i) \log \left( \sum_{a=1}^{T} P(\tau_a|\theta_i) P(w|\tau_a) \right) \qquad (1)$$

In this equation $V$ is the set of all the words in the vocabulary, $c(w, r_i)$ is the count of word $w$ in the expertise document $r_i$ and $p(\tau_a|\theta_i)$ is the probability of selection of the topic $\tau_a$ for document $r_i$. We can use the EM algorithm to compute the maximum likelihood estimate of all parameters including $p(a|\theta_i)$ and $p(w|\tau_a)$. After learning all the parameters, we can represent each expert $e_i$ using the topic vector $\tau_i(\tau_{1i}, \dots \tau_{Ti})$. Furthermore, using the estimated values of $p(w|\tau_a)$, we can infer the topic representation for each paper $p_j$ as $\tau'_j(\tau_{1j}, \dots \tau_{Ti})$. After representing papers and reviewers using the above topic model, now we can describe the matching algorithm for retrieving the *k*-top reviewers for each paper.

Following the idea of uncapacitated facility location analysis (UFLA), our intuition to match each paper $p_j$ with a set of $k$ reviewers can be explained like follows. We can imagine each relevant topic $t_a$ of paper $p_j$ as a consumer with demand of $\tau_{aj}$ and each available expert $e_i$ as a candidate facility location. According to the condition $C_1$, We should open $k$ facilities (i.e. select $k$ reviewers) among $M$ candidate facility locations (i.e. the number of all available reviewers) to serve the $T$ customers (i.e. the number of aspects of paper $p_j$) such that the overall cost of opening those connection be minimal.

In order to match the paper $p_j$ with $M$ available reviewers, there are $T$ customers and $M$ candidate facility locations. As mentioned before, the objective function in UFLA is composed of two parts: 1) *Building Cost:* which indicates the cost of opening the facility at a specific location (i.e. the cost of selection of expert $e_i$ for paper $p_j$) and 2) *Communication Cost:* which indicates the access cost of a customer (i.e. an aspect/topic) to the nearest facility location (i.e. the best reviewer for that specific aspect).

To maximize the aspect coverage of the assigned group (i.e. to satisfy the condition $C_2$), the assigned group should be selected such that each topic $\tau_a$(i.e. the a$^{th}$ customer) of paper $p_j$ can be assigned to a near facility location (i.e. to a reviewer who is able to cover topic $a$). On the other hand, to maximize the confidence of each assigned reviewers (i.e. to satisfy the condition $C_3$); we should select low-cost facility locations (i.e. reviewers with maximum confidence).

Therefore, we can define the building cost and communication cost in our framework as follows:

1- *Building cost* of assignment of reviewer $e_i$ to paper $p_j$: $cost(e_i) = D(\overrightarrow{e_i}||\overrightarrow{p_j})$, where $\overrightarrow{e_i}$ and $\overrightarrow{p_j}$ are the topic vectors of reviewer $e_i$ and paper $p_j$ and $D(\overrightarrow{e_i}||\overrightarrow{p_j})$ indicates the Kullback-Leibler (KL) divergence value of these vectors.

2- *Communication cost* of assignment of aspect $a$ of paper $p_j$ to reviewer $e_i$: $cost(e_i, \tau_{aj}) = D(\overrightarrow{e_i}||\overrightarrow{v_a})$, where $\overrightarrow{e_i}$ is the topic vector of reviewer $e_i$ and $\overrightarrow{v_a}$ indicated the unit vector with all zero elements except for topic $a$.

Intuitively, if the distribution of vectors $\overrightarrow{e_i}$ and $\overrightarrow{p_j}$ is very similar to each other, then we expect that they might be related to the same topics and also their KL divergence value will be very small. As a result, the facility (i.e. reviewer) $e_i$ will be a very low building cost facility for paper $p_j$. On the other hand, if reviewer $e_i$ has the skill $a$, we expect that the weight of his/her corresponding element in vector $\overrightarrow{e_i}$ might be higher in comparison with other elements and accordingly the communication cost of customer $a$ (i.e. topic a) and facility $r_i$ (i.e. reviewer) will be low.

To sum up, the objective function of unconstraint expert matching problem can be represented as follows:

$$cost(S, p_j) = \lambda \sum_{i=1}^{k} D(e_i||p_j) + (1-\lambda) \sum_{a=1}^{T} \tau_{aj} \min_{s \in S} D(s||v_a)$$

Where $S$ is the set of selected reviewers for paper $p_j$ and $\tau_{aj}$ indicates the weight of topic $a$ in topic vector of paper $p_j$. To optimize the above objective function we use the local greedy search method introduced in algorithm 1. The output of the algorithm 1 is the *k*-best reviewers for paper $p_j$ which have not only the maximum confidence but also collectively can cover all aspects of that paper.

## 3.2 Explicit Aspects- Constraint matching

The second problem of expert matching (i.e. *Explicit Aspects-Constraint matching)* concerns with the assignment of $N$ papers to $M$ reviewers such that in addition to the conditions $C_1$, $C_2$ and $C_3$, the following condition is satisfied:

**$C_4$**–(Capacity Condition) each reviewer has a limited capacity and can only be assigned to a limited and predefined number of papers.

In contrast with the first problem of expert matching, in this problem, we assume that the aspects/skills of the papers and experts are explicitly determined. This is a valid assumption because in many applications, the required skills of a project and also the related skills of an expert can be explicitly described by some few keywords.

The constraint **$C_4$** makes the matching problem very hard, indeed this matching problem is also NP-hard; furthermore, in this problem the conditions **$C_2$** and **$C_3$** are in competition with each other. In other words, maximizing the global average confidence of the assigned group may result in formation of the non-optimal converging groups. In this section, we formally model this problem using Capacitated Facility location analysis (CFLA) and also propose an exact linear programming solution for it.

Similar to the UFLA method, in the CFLA framework of constraint expert matching, each aspect/topic of paper $p_j$ is considered as a customer and each reviewer/expert is considered as a candidate facility location. In order to form optimal aspect covering groups, each required aspect of a paper should be assigned to reviewer who is able to cover that topic. On the other hand, it is preferable that the assigned reviewers of a paper be able to cover all required skills of that paper. In CFLA framework, the first condition can be satisfied by modeling the communication cost between aspects and reviewers and the second condition can be satisfied by modeling the building cost of each reviewer.

Algorithm 2 indicates the linear programming solution[1] for this CFLA problem. In this linear program, matrix $M_{ij}$ is a $(N \times M)$ binary decision matrix that its element indicates the assignment of papers to the reviewers. Specifically, element $m_{ij} = 1$ if and only if in the final solution, the paper $p_i$ is assigned to the reviewer $r_j$. As a binary decision variable, $X(i, j, t)$ indicates the assignment of topic $t$ of paper $p_i$ (i.e. a customer) to the reviewer $e_j$ (i.e. a facility). Finally, $A(N \times T)$ is the paper-topic association matrix; where $A(i, t) = 1$ if and only if paper $p_i$ is related to the topic $t$.

In this linear program, constraint $T_1$ shows that sum of elements of each row in $M_{ij}$ should be equal to $k$; this means that each paper should be assigned exactly to $k$ reviewers. Constraint $T_2$ indicates that the sum of elements of each column of $M_{ij}$ should be less than $c$ (i.e. capacity of reviewers); this means that each

reviewer can only be assigned to at most $c$ papers. Constraint $T_3$ indicates that for each related topics of paper $p_i$ (i.e. for topics that $A(i,t)=1$) at least one reviewer should be assigned. As an important constraint $T_4$ indicates that topic t of paper $p_i$ can be assigned to the reviewer $r_j$ only if paper $p_i$ is assigned to the reviewer $r_j$. In other words, if decision variable $X(i,j,t)=1$ then the value of $M_{ij}$ should be equal to 1; this means we can assign topic $t$ of paper $p_i$ to reviewer $r_j$ only if $r_j$ is selected for paper $p_i$.

It is worth mentioning that the objective function in Algorithm 2 is same as the objective function of equation (1) with this difference that in algorithm 2, it is defined to globally optimize the matching of all papers (i.e. the outer sum is defined on all papers). As another point, the minimum distance in equation (1) is replaced by the decision variable $X(i,j,k)$. Intuitively, by minimizing the objective function two conditions of the problem can be satisfied. Firstly, paper $p_i$ is assigned to reviewer $r_j$ (i.e. $M_{ij}=1$) when the building cost (i.e. $BCost(i,j)$) of this selection is low; this part of objective function can satisfy confidence maximization. On the other hand, minimization of the communication cost of topic $t$ (i.e. $CCost(j,t)$) results in the assignment of topic $t$ to reviewer $r_j$ such that $r_j$ is able to cover this topic; this part of the objective function can satisfy the coverage maximization condition and parameter $\lambda$ can be used to make the tradeoff between confidence and coverage conditions.

**Algorithm 2. Linear programming solution of CFLA**

$$Opt = \min\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\left(\lambda\, M_{ij} BCost(i,j) + (1-\lambda)\sum_{t=1}^{l} CCost(j,t)X(i,j,t)\right)\right)$$

$$T_1 - \forall i: \sum_{j=1}^{M} M_{ij} = k \qquad T_4 - \forall i,j,t: X(i,j,t) \le M_{ij}$$

$$T_2 - \forall j: \sum_{i=1}^{N} M_{ij} \le c \qquad T_5 - \forall i,j: M_{ij} \in \{0,1\}$$

$$T_3 - \forall i,t: A(i,t) \le \sum_{j=1}^{m} X(i,j,t) \qquad T_6 - \forall i,j,k: X(i,j,k) \in \{0,1\}$$

To optimize the coverage and confidence of the assigned groups, we can define the building and communication cost as follow:

$$BCost(i,j) = \frac{aspect(p_i) \cap aspect(r_j)}{aspect(p_i)}$$

$$CCost(j,t) = \begin{cases} 0 \; if \; t \in aspect(r_j) \\ 1 \; otherwise \end{cases}$$

## 3.3 Implicit Aspects- Constraint matching

The constraints in the third problem of expert matching (i.e. *Implicit Aspects- Constraint matching)* are similar to the second problem but in this problem the topics/aspects of papers and reviewers are not predetermined. We use the PLSA topic modeling to infer the topic representation for each paper and reviewer. Utilizing the inferred topic vectors, we can use algorithm 2 for expert matching. Similar to the proposed method for estimation of Building and Communication cost in problem 1, we define the building and communication cost in the same way.

## 4. Related work

The problem of expert group formation has recently attracted a lot of attention in information retrieval [2] [3] [8] and social network communities [9]. This problem can be considered as an extension of the expert finding[10] problem. Expert finding is a well studied problem in IR community which concerns itself with the finding of knowledgeable people in a given topic[11]. Several algorithms are proposed for expert finding problem including the language modeling[11], voting model, and person centric language modeling [12]. While initial approaches for expert finding concern with finding the knowledgeable persons in an organization[11] , recent methods focused on finding experts in the bibliographic data[13].

The problem of multi aspect expert group formation is initially introduced by Karimzadegan and Zhai[2]. Specially, they considered the first problem of expert matching (i.e. Implicit Aspects- unconstrained matching) and proposed three different strategies to find a group of experts that maximally cover all required skills of a given query. The proposed methods [2] are redundancy removal, expert aspect modeling and query aspect modeling.

The idea of the redundancy removal method is to diversify the set of retrieved experts such that experts with various skills can be selected to cover all required skill of the query. In the query aspect modeling method, a multi-aspect query is segmented into semantically diverse parts such that each part can be considered as a single aspect query; then, each query part is used to retrieve relevant experts. Then, the union of retrieved experts is considered as the final answer.

The most effective proposed method in[2] is the expert aspect modeling. Similar to our approach for the first problem of expert matching, it is based on learning a topic vector representation for experts and queries (i.e. in paper-review assignment problem, each query is equivalent to a paper). Using these vector topics, the *Next Best greedy* approach is utilized to form the optimal skill covering group. In this approach, the members of a group are selected step by step; At step $k$, an expert (i.e. *the best candidate in step k*) which minimizes the following objective function is selected as a new member of the group:

$$D(\theta_q || \left(\frac{\sigma}{k-1}\sum_{i=1}^{k-1} p(a|\theta_i) + (1-\sigma)p(a|\theta_k)\right))$$

In this equation, $p(a|\theta_i)$ indicates the topic distribution of the i[th] selected member of the group, the probability $p(a|\theta_k)$ is the topic distribution of the k[th] candidate expert and $\sigma$ is a parameter to model the redundancy of skills in selected members. Thus, the above objective function is the KL divergence of the topic distribution of the query/paper and the resulting group after selection of the k-th expert candidate.

In contrast with the *Next Best* greedy approach [2], our proposed method for implicit aspect matching problem (illustrated in Algorithm 1), measures at each iteration the ability of whole *k*-members of a group to cover the required skills of the query and as a result, if a non-appropriate member is selected at step $k$ it can be eliminated at next steps. However, in the *Next Best* greedy approach, a non-appropriate selected member at step $k$ cannot be changed. On the other hand, the *Next Best* greedy approach cannot easily be extended for constraint matching problems (i.e. problem 2 and 3). In contrast, our FLA framework for expert matching can be easily extended for constraint and explicit aspect matching problems.

The problem of constraint expert group formation (i.e. Problem 2 and 3) recently introduced in [3], can be considered as an extension of the paper-review assignment problem [14][15]. While these initial approaches for this problem concern with the

assignment of papers to relevant reviewers, Karmizadegan and Zhai [3], introduced the problem of multi-aspect constraint paper review matching (i.e. problem 2 and 3). They proposed a heuristic method based on the integer linear programming (ILP)
measure, it is not able to optimize the coverage measure. On the other hand, their proposed method for implicit topics/aspects (problem 3) has very low coverage and confidence in comparison with our CFLA method. We use the methods proposed in [3] and [2] as our baseline model and also use the same dataset to make the results comparable.

Recently, Tang et al. [8] proposed a general framework based on the convex cost flow optimization for expert matching. Their proposed method mainly focused on authority and soft load balancing constraints and does not solve the coverage and confidence conditions optimally. As another related line of research, authors of [9] proposed a method to find a group of experts in a social network. Although this problem is closely related to the expert matching problem, their main concern is to find a group of experts in a social network which are able to contribute to each other easily.

## 5. Experiments

In this section, we present the test data and measures used for evaluating our methods.

### 5.1 Data set

We used the dataset introduced in [2] to evaluate our proposed methods. This dataset is used in several research papers ([2] [3] [8]) and to the best of our knowledge is the only available dataset for multi aspect team formation problem. The dataset is crawled from the abstract papers of ACM SIGIR proceedings from years 1971-2006. Authors of these papers are considered as the prospective reviewers/experts. For modeling reviewers' expertise, a profile is created for each author by concatenation of all papers written by that specific author. The SIGIR 2007 papers are used to simulate papers that are to be reviewed. In this dataset, there are 73 papers with at least two aspects. A gold standard is created for this dataset by identifying 25 major subtopics for these papers and then assignment of subtopics to all papers and the reviewers by a human expert. In total, there are 73 papers and 189 reviewers in this dataset which is publicly available at http://timan.cs.uiuc.edu/data/review.html.

### 5.2 Evaluation measures

While the multi-aspect team formation problem can be cast as a retrieval problem, the traditional relevance-based precision and recall measures cannot be directly applied to measure performance, because they are unable to reflect the coverage and confidence measures in the assigned groups. To measure the performance of our multi-aspect matching algorithms, we used the *Coverage* and *Average Confidence* measures proposed in [2].

Coverage score measures the number of different distinct topic aspects that are covered by the $k$ assigned reviewers as a function of aspects in the query. Consider a paper with $n_A$ topic aspects $A_1, \ldots, A_{n_A}$ and let $n_r$ denote the number of distinct topic aspects that the $k$ assigned reviewers can cover. Coverage can be defined as the percentage of topic aspects covered by these reviewers:

$$\text{Coverage} = \frac{n_r}{n_A}$$

As mentioned before, in addition to the maximizing the coverage of topic aspects, the assignments that maximize the confidence of the assigned reviewers are more preferable. Specifically, in the same level of coverage, we would prefer an assignment where each reviewer is able to cover as many aspects as possible. Using the notations introduced earlier, the Average Confidence measure is defined as follows:

$$\text{Average Confidence} = \frac{\sum_{i=1}^{n_A} \frac{n_{A_i}}{k}}{n_A}$$

In this equation, $k$ is the number of assigned reviewers and $n_{A_i}$ indicates the number of assigned reviewers that can cover the $i^{th}$ topic/aspect.

### 5.3 Baseline Methods

In the experimental result section, the FLA framework of multi-aspect expert matching is compared with the methods proposed in [2] and [3]. As another baseline, we compare the result of FLA for the first problem of expert matching (i.e. *Implicit Aspects-Unconstraint matching*) with a standard retrieval model (i.e. language modeling with Dirichlet smoothing). In this method, for each query/paper, expertise documents of reviewers are ranked according to language model score and then the top $k$ reviewers are selected as the assigned expert group for that specific paper. In comparison of the proposed models, statistically significant improvements are measured using a Wilcoxon Signed-Ranktest at the level of 0.05.

## 6. Experimental Results

In this section, an extensive set of experiments were conducted to address the following questions:

1) In the first problem of expert matching (i.e. *Implicit aspects-unconstraint matching*), how good is the performance of the UFLA approach? In section 6.1, we compare the performance of UFLA with various baselines proposed in [2]. In particular, we compare two greedy approaches for expert matching namely, *Next Best* search and *Local Search* strategies.
2) What is the impact of building and communication cost on the coverage and confidence measures in our FLA framework? How good is the performance of the FLA framework for different values of the parameter λ?
3) How good is the performance of the proposed framework for constraint expert matching problems in comparison with the heuristic methods proposed in [3] ?

### 6.1 Implicit Aspects- Unconstraint matching

In this section, we compare the UFLA method described in section 3.1 with the language Model (LM), Redundancy Removal (RR), and the *Next Best* greedy method described in related work section. In order to evaluate the effectiveness of our methods, we compare all well-tuned methods. In these experiments, 73 papers are assigned to the 189 available reviewers such that each paper gets exactly 3 reviewers. Table 2 indicates the coverage and the average confidence scores and the percentage of improvement for UFLA method.

**Table 2. Comparison of the UFLA method with baseline algorithms for the first problem of expert matching- statically significant improvement is shown by * symbol.**

| Measure | Coverage | | Average Confidence | |
|---|---|---|---|---|
| | result | %Δ v.s. UFLA | result | %Δ v.s. UFLA |
| Baseline-LM | 0.750 | +20.0% | 0.420 | +34.3% |
| Baseline-RR | 0.770 | +16.9% | 0.450 | +25.3% |
| Baseline-*Next Best* | 0.869 | +3.6% | 0.501 | +12.6% |
| UFLA | **0.900*** | - | **0.564*** | - |

According to the Table 2, the performance of author topic modeling methods (i.e. *Next Best* and UFLA) are better than other baseline methods and also the coverage and especially the average confidence of the UFLA method is better than the *Next best* search method.

Since the performance of the *Next Best* and the *UFLA* methods are dependent on the quality of the topic learning model, in order to fairly compare these methods, we use another model to learn these topics. In this model, we use the skills/aspects associated with each reviewer from the golden set (i.e. in equation (1), the parameters $p(a|\theta_i)$ are known) and just learn the word distributions for each topic (i.e. the only unknown parameters in equation (1) are the $p(w|a)$). Using the estimated word distribution parameters, we infer the topic vector for each paper and run the *Next Best* and the UFLA algorithms in the same manner using the new topic vectors. Table 3 indicates the coverage and average confidence for this experiment.

**Table 3. Comparison of the UFLA and *Next Best* methods- for the improved topic learning model- statically significant improvement is shown by * symbol.**

| Measure | Coverage | | Average Confidence | |
|---|---|---|---|---|
| | result | %Δ v.s. UFLA | result | %Δ v.s. UFLA |
| Baseline-*Next Best* | 0.890 | +7.1% | 0.660 | +3.0% |
| UFLA | **0.953*** | - | **0.680** | - |

According to tabel3, by improving the topic modeling, the coverage and average confidence of both FLA and *Next Best* methods are improved. However, the performance of UFLA is again better than the *Next Best* greedy matching. The result of these experiments (i.e. Table 2 and Table 3) indicate that independent of the method used for topic learning, the performance of UFLA matching is always better than the *Next Best* method for expert matching.

To better understand the behavior of *Next Best* and the UFLA methods, we examine the impact of $\sigma$ and $\lambda$ on performance of these methods. As mentioned before, the parameter $\sigma$ in *Next Best* method models the skill redundancy in the assigned groups and the parameter $\lambda$ makes the balance between building cost and commutation cost in the UFLA framework. Figure 3 and Figure 4, indicates the sensitivity of the coverage and the average confidence measures on these parameters for the UFLA and *Next Best* algorithms.

According to Figure 3, while the coverage score of the *Next Best* method fluctuates for different values of $\sigma$, the coverage of UFLA method is stable for $\lambda > 0$. This experiment also shows that eliminating the building cost from the objective function (i.e. $\lambda = 0$ in equation (1)) significantly reduces the performance of UFLA matching algorithm. The same pattern is observable in Figure 4.

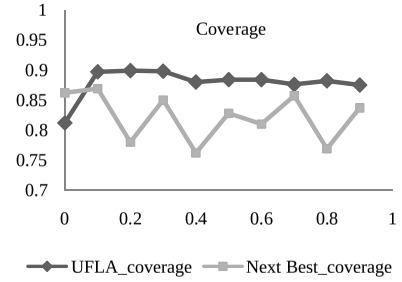



**Figure 3. The sensitivity of coverage measure on parameter λ and σ**

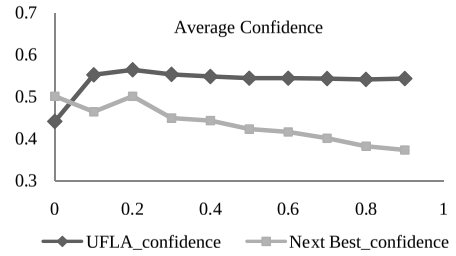



**Figure 4. The sensitivity of confidence measure on parameter λ and σ**

## 6.2 Explicit Aspects- Constraint Matching

In this section, we compare our CFLA method for constraint matching with the baseline methods proposed in [3]. The first baseline algorithm is the greedy approach proposed for constraint expert matching proposed in [3]. In this method, First, the papers are decreasingly sorted according to the number of subtopics they contain, i.e., the paper with the largest number of subtopics is ranked first. Then start off with this ranked list of the papers. At each assignment stage, the best reviewer that can cover most subtopics of the paper is assigned. In addition, the review quota and paper quota are checked, i.e., the number of papers assigned to each reviewer and the number of reviewers assigned to each paper. If the review quota is reached, that reviewer is removed from our reviewer pool; the same is done when the paper quota is satisfied. This process is repeated until reviewers are assigned to all the papers. The second baseline algorithm is the integer linear programming proposed in [3] to match papers with reviewer. This method tries to globally maximize the number of covered aspects of the assigned groups.

Before comparison with the baseline algorithms, we examine the impact of building and communication cost in CFLA model on coverage and average confidence measures. Figure 5, indicates the sensitivity of coverage for different size of program committee

(i.e. number of available reviewers). In these experiments, each paper is assigned to 3 reviewers and the capacity of each reviewer is equal to 5. For each program committee size, we randomly select specified number of experts from all available experts (i.e. 189 experts) and repeat each experiment 10 times and report the average of coverage in Figure 5.

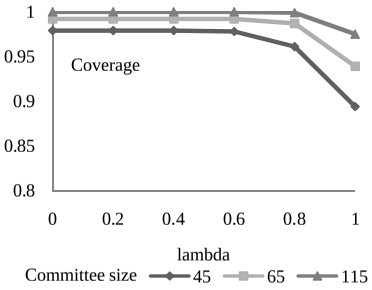



**Figure 5. The sensitivity of average confidence on parameter λ CFLA matching- each date series indicates coverage score for different program committee size.**

According to the Figure 5, we can see that increasing the size of committee (i.e. available experts) improves the coverage score and on the other hand increasing the parameter $\lambda$ (i.e. increasing the building cost and decreasing the communication cost in objective function of the CFLA) decreases the coverage score of the assigned groups. This experiment shows that by emphasizing on communication cost in the objective function of CFLA, the coverage score of assigned groups can be increased.

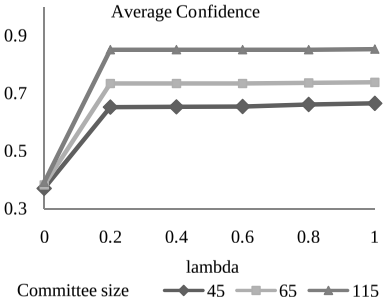



**Figure 6. The sensitivity of average confidence on parameter $\lambda$ CFLA matching- each date series indicates coverage score for different program committee size.**

Figure 6 indicates the result of above mentioned experiment in terms of the average confidence score. While the average confidence is stable for $\lambda > 0$, the maximum and minimum values for $\lambda$ is occurred at $\lambda = 1$ and $\lambda = 0$ respectivle. Specifically, for $\lambda = 0$, the value of average confidence score is reduced substantially which means that by ignoring the building cost, the average confidence score reduces. This experiment shows that the coverage and the average confidence scores are contradicting constraints. In all other experiments of this section, we use $\lambda = 0.5$ to make a balance between the coverage and the average confidence measures.

In the next experiment, we compare the proposed CFLA method with the integer linear programming method [3] and the greedy matching algorithms for different program committee sizes (i.e. number of available reviewers). Each experiment is repeated 10 times and the average of scores are reported. In this experiment, each paper is assigned to 3 reviewers and the capacity of each reviewer is 5. Figure 7 indicates the coverage score of CFLA (i.e. proposed model), ILP and the greedy approach for different sizes of program committee.

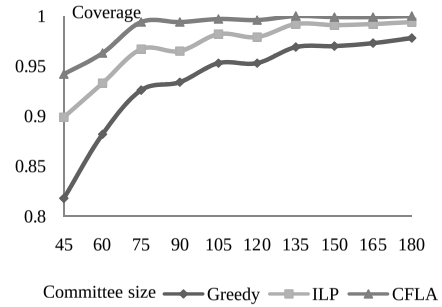



**Figure 7. Coverage score of Greedy, ILP and CFLA for various committee program sizes.**

According to Figure 7, by increasing the size of program committee the coverage score is increasing for all methods. In addition, for all committee program sizes, the coverage score of the CFLA method is always better than the greedy and ILP methods. Specifically, the CFLA method can significantly improve the coverage score for small program committee sizes (i.e. less than 120 available reviewers). Table 4 indicates the average confidence score of the CFLA (i.e. proposed model), ILP and the greedy approach for this experiment.

**Table 4. Comparison of all method based on the Average Confidence**

| Committee size | 45 | 55 | 65 | 105 | 185 |
|---|---|---|---|---|---|
| Greedy | 0.550 | 0.634 | 0.665 | 0.798 | 0.882 |
| ILP | **0.651** | 0.708 | 0.724 | 0.831 | 0.914 |
| CFLA | 0.647 | **0.710** | **0.729** | **0.837** | **0.916** |

According to Table 4, for all matching methods, by increasing the size of committee (i.e. increasing the size of available experts), the average confidence score is improved. The performance of ILP and CFLA are almost the same and both are better than tge greedy method. According to this experiment, the CFLA method can detect expert groups with significantly better coverage score in comparison with the ILP method without reduction of the average confidence score. Specifically, it can improve the coverage score up to 8.90% for small committee sizes, while the variation of the average confidence score is negligible.

In the next experiment, we fix the number of reviewers to 30, and vary the number of papers each reviewer can review. In order to avoid bias, we repeat the sampling process (selection 30 reviewers) for 10 times and get the average. The coverage scores are shown in Figure 8.

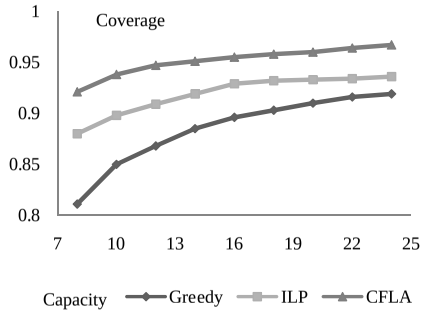



**Figure 8. Coverage score of Greedy, ILP and CFLA for different capacity of reviewers.**

As we increase the number of papers that each reviewer can get, we are also increasing the resources, and as a result, the performance of all algorithms becomes better. Also, comparing the CFLA method with the ILP and greedy approach, the performance of the CFLA method is significantly better than the greedy and ILP method for all values of capacity of reviewers. Table 5 indicates the average confidence scores for this experiment.

**Table 5. Comparison of all method based on the Coverage and Average Confidence**

| Capacity/Size | 8 | 12 | 16 | 20 | 24 |
|---|---|---|---|---|---|
| Greedy | 0.526 | 0.6 | 0.629 | 0.647 | 0.658 |
| ILP | 0.614 | 0.64 | 0.654 | 0.664 | 0.668 |
| CFLA | 0.615 | 0.64 | 0.655 | 0.664 | 0.668 |

The average confidence score of the CFLA and the ILP methods are almost the same but both are better than the greedy algorithm. This experiment also indicates that the CFLA algorithm can improve the coverage measure while retain the average confidence in the same level. It means that the CFLA can better distribute papers among available reviewers.

In the last experiment, we compare the performance of CFLA and ILP when very limited recourse (i.e. reviewers) is available. In this experiment, the maximum number of reviewers is 10 for 73 papers. Again we randomly select 10 reviewers and we repeat the sampling process for 10 times and get the average. Each paper gets three reviewers and the number of papers that each reviewer can get is calculated according to the number of reviewers that we have. For example, if we have five reviewers, each should get 44 papers. Figure 9 indicates the coverage measure of CFLA and ILP methods.

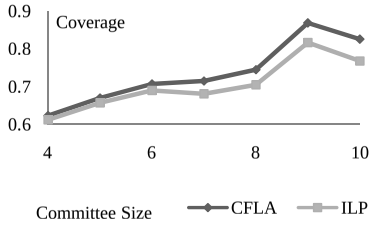



**Figure 9. Coverage score of ILP and CFLA for very limited resources**

The result of average confidence is also reported in Table 6. This experiments shows that for very limited recourses the quality of matching for CFLA is better than the ILP in terms of both the coverage and average confidence.

**Table 6. Average Confidence score of ILP and CFLA for very limited resources**

| Capacity/Size | 4 | 6 | 8 | 10 |
|---|---|---|---|---|
| ILP | 0.243 | 0.274 | 0.289 | 0.318 |
| CFLA | **0.270*** | **0.307*** | **0.355*** | **0.441*** |

## 6.3 Implicit Aspects- constraint Matching

In this section, we examine the quality of matching experts for third problem of expert matching. In this case, the aspects/skills of papers and reviewers are implicitly given in abstract and expertise documents. Similar to previous experiments, we use $\lambda = 0.5$ to make a balance between building and communication cost.

While our CFLA method can be directly applied to the probabilistic assignments of subtopics given by PLSA, intuitively, not all the predictions are reliable, especially the low-probability ones. Thus we experimented with pruning low probability values learned with PLSA (i.e., setting low topic probability elements to zero). The greedy approach is not applicable for this matching problem because the aspects/topics of papers and reviewers are not predetermined. So, we use the ILP method introduced in[3] as our baseline model. Table 7 indicates the result of matching for the best parameters (i.e. cut-off =3). In this experiment, the size of the program committee size is 189 and the capacity of each reviewer is 5.

**Table 7. Comparison of ILP and CFLA for implicit Aspect**

| | Coverage | | Average Confidence | |
|---|---|---|---|---|
| | Value | %**Δ** v.s ILP | Value | %**Δ** v.s ILP |
| ILP | 0.715 | - | 0.347 | - |
| CFLA | **0.828*** | +15.8% | **0.436*** | +25.6% |

Figure 10 indicates the effect of different values for cut-off and sensitivity of algorithms to parameter $\lambda$ on coverage score. In this figure, each data series indicate a method and a value of cut-off for example, CFLA (5) indicate expert matching using CFLA method and setting the cut-off value equals to 5. i.e. only top 5 topic is used in topic vector of reviewers and papers.

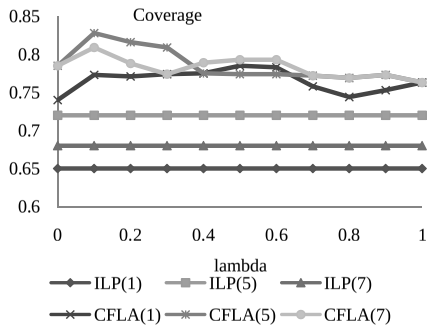



**Figure 10. Coverage score of ILP and CFLA for implicit aspects**

According to Figure 10, setting the cut-off equals 1 reduces the performance of both algorithms. Also, setting the cut-off more than five increases the noise and as a result the coverage reduces. The optimal value for cut-off is near to average number of required skills for each paper in the golden measure (i.e. 5 aspects for each paper). Although, the coverage of the CFLA method fluctuates for different values of $\lambda$, for all values the coverage is significantly better than the ILP model. To sum up, according to this experiment, the performance of the CFLA method is significantly better than the ILP method in both coverage and average confidence measures for implicit aspect-constraint matching problem.

## 7. Conclusion

In real scenarios, several and sometimes diverse skills are needed to perform a project successfully and completely. In this paper, we consider the problem of expert group formation (i.e. expert matching) to optimally assign a set of available experts to a project. Three types of group formation problems are considered in this paper. In the first problem, we assume that the required skills of a project and also the relevant skills of experts are implicitly expressed by the text documents. The second problem concerns with the assignment of experts to multiple projects such that each expert should be involved in a limited number of projects and the third problem is the combination of the first and the second problems. The assigned group of experts to each project should be able to cover all required skills of that project and preferably, each member of a assigned group should also be able to cover all the these aspects. A unified framework based on the facility location analysis is proposed in this paper to address these problems. As a case study, we consider the problem of multi-aspect review assignment which is a common task in conference and journal organizations. Several experiments are conducted on a real dataset to compare the performance of the proposed framework with the state-of-the-art methods. Our experiments show that the FLA framework can significantly improve the performance of expert matching in terms of two performance measures.

This work was in part supported by a grant from *Iran telecommunication research center* (ITRC). We also would like to thank the authors of [3] for making their test collection publicly available.

## 8. References

1 Gonzalez, Teofilo F. *Handbook of Approximation Algorithms and Metaheuristics*. Chapman & Hall/Crc Computer & Information Science Series, 2007.

2 Karimzadehgan, Maryam, Zhai, ChengXiang, and Belford, Geneva. Multi-aspect expertise matching for review assignment. In *Proceedings of the 17th ACM conference on Information and knowledge management* ( 2008), 1113-1122.

3 Karimzadehgan, Maryam and Zhai, ChengXiang. Integer linear programming for Constrained Multi-Aspect Committee Review Assignment. *Inf. Process. Manage.*, 48 (2012), 725--740.

4 Gabor, A.F. and Ommeren van, J.C.W. *Approximation algorithms for facility location problems with discrete subadditive cost functions*. Department of Applied Mathematics, University of Twente, Enschede, 2005.

5 Chudak, Fabián A. and Williamson, David P. Improved approximation algorithms for capacitated facility location problems. *Mathematical Programming: Series A and B*, 102, 2 (2005), 207 - 222.

6 Charikar, Moses and Guha, Sudipto. Improved Combinatorial Algorithms for the Facility Location and k-Median Problems. In *FOCS '99 Proceedings of the 40th Annual Symposium on Foundations of Computer Science* ( 1999), 378.

7 Hofmann, Thomas. Probabilistic latent semantic indexing. In *Proceedings of ACMSIGIR'99* ( 1999), 50-57.

8 Tang, Wenbin, Tang, Jie, Lei, Tao, Tan, Chenhao Gao, Bo, and Li, Tian. On optimization of expertise matching with various constraints. *Neurocomput.*, 76, 1 (2012), 71-83.

9 Lappas, Theodoros, Liu, Kun, and Terzi, Evimaria. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data minig* ( 2009), 467-476.

10 Balog, K, Soboroff, I, Thomas, P, Craswell, N, and Bailey, P. The Seventeenth Text Retrieval Conference Proceedings (TREC 2008). In *NIST* ( 2009).

11 Balog, Krisztian, Azzopardi, Leif, and de Rijke, Maarten. A language modeling framework for expert finding. *Inf. Process. Manage.*, 45, 1 (2009), 1-19.

12 Serdyukov, Pavel and Hiemstra, Djoerd. Modeling documents as mixtures of persons for expert finding. In *Proceedings of the IR research, 30th European conference on Advances in information retrieval* ( 2008), 309-320.

13 Deng, Hongbo, Han, Jiawei, Michael, Lyu, and Irwin, King. Modeling and Exploiting Heterogeneous Bibliographic Networks for Expertise Ranking. In *2012 ACM/IEEE Joint Conference on Digital Libraries (JCDL 2012)* ( 2012).

14 Taylor, Camillo J. *On the optimal assignment of conference papers to reviewers*. University of Pennsylvania.Technical Reports, 2009.

15 Mimno, David and McCallum, Andrew. Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* ( 2007), 500-509.

# An Ontology Derived from Heterogeneous Sustainability Indicator Set Documents

Lida Ghahremanloo[1], James A. Thom[1], Liam Magee[2]
[1]School of Computer Science and Information Technology
[2]School of Global Studies, Social Science and Planning
RMIT University
Melbourne, Australia
(lida.ghahremanloo,james.thom,liam.magee)@rmit.edu.au

## ABSTRACT

We present an *ontology* to represent the key concepts of sustainability indicators that are increasingly being used to measure the economic, environmental and social properties of complex systems. There have been few efforts to represent multiple indicators formally, in spite of the fact that comparison of indicators and measurements across reporting contexts is a critical task. In this paper, we apply the METHONTOLOGY approach to guide the construction of two design candidates we term *Generic* and *Specific*. Of the two, the generic design is more abstract, with fewer classes and properties. Documents describing two indicator systems − the Global Reporting Initiative and the Organisation for Economic Co-operation and Development − are used in the design of both candidate ontologies. We then evaluate both ontology designs using the ROMEO approach, to calculate their level of coverage against the seen indicators, as well as against an unseen third indicator set (the United Nations Statistics Division). We also show that use of existing structured approaches like METHONTOLOGY and ROMEO can reduce ambiguity in ontology design and evaluation for domain-level ontologies. It is concluded that where an ontology needs to be designed for both seen and unseen indicator systems, a generic and reusable design is preferable.

## Categories and Subject Descriptors

H.2 [**Database Management**]: Logical Design—*Data models*; I.2 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Representations (procedural and rule-based)*; D.2 [**Software Engineering**]: Interoperability

## Keywords

Ontology Engineering, Ontology Evaluation, Sustainability Indicators


*ADCS '12* December 05 - 06 2012, Dunedin, New Zealand


## 1. INTRODUCTION

Since the publication of the Club of Rome's "The Limits to Growth" in the early 1970s, the sustainability of natural and social systems has become a pressing concern. This problem has become increasingly urgent, given the depletion of our natural stocks due to past and present economic development. To respond to such a challenge, it is vitally important to develop reliable and robust *indicators*, to measure the sustainability of our complex economic, environmental and social systems. These provide the tools to manage current and future development responsibly. In response, a number of indicator systems have been developed and are in use today. These include generalized reporting standards, such as those developed by the Global Reporting Initiative (GRI)[1], the Organization for Economic Co-operation and Development (OECD)[2] and the United Nations Statistics Division (UN Social Indicators)[3].

The number of indicator systems itself poses problems for would-be reporting organisations – how to find which indicators best represent the challenges of their specific context? To date, there have been few efforts to represent *multiple* indicator systems in a systematic way. In particular, we see considerable applications for the representation of indicators in a formal *ontology*. In computing disciplines, an ontology refers to a formal explicit specification of a shared conceptualisation for a domain of interests consisting of a set of *concepts*, *relationships* and *individuals* that are reused within enterprises [13]. Through use of such a formal construct, it is possible to develop a consistent definition of what an indicator is, and how it can be applied. This in turn would allow organisations to browse and review different kinds of indicators for different measurement applications, and to enable some degree of comparison and bench-marking between them. A further challenge exists in the design and evaluation of domain-level ontologies. Although in ontology research several approaches have been proposed for structuring the knowledge into different levels of abstraction [7, 21, 24], there is still considerable ambiguity involved in the processes of construction and evaluation. Several promising approaches, such as METHONTOLOGY [8] and ROMEO [25], have been developed to provide clear structural guidelines, in order to simplify the complexities and eliminate doubts in ontology design and evaluation. However these have not yet been deployed in the context of sustainability

[1]http://www.globalreporting.org/Home
[2]http://www.oecd.org/home
[3]http://www.un.org/esa

indicators. The current study showcases an effort to build a robust and reusable ontology for sustainability indicators, by incorporating these approaches into an integrated framework for design and evaluation.

In this paper, we present an *ontology for sustainability indicator sets (OSIS)*, for representing information about sustainability indicator systems. In Section 2, we first review the METHONTOLOGY ontology engineering and the ROMEO ontology evaluation approaches. Section 3 discusses the development and support activities given from METHONTOLOGY to design two candidates for OSIS. This is followed in Section 4 by the evaluation activity using ROMEO to validate two ontology models with a set of experiments. Section 5 presents the results and our findings from performing the experiments. Lastly, Section 6 concludes this work and discusses future work to pursue.

## 2. RELATED WORK

Domain-level ontologies have been developed in a wide range of disciplinary areas, with bioinformatics and the life sciences being particularly prominent examples. Relatively little attention has been devoted to establishing ontologies in the domain of sustainability science. Kumazawa et al. [16] demonstrate one such example, using a problem-based approach to structure a complex conceptual hierarchy in support of a general knowledge base. Similarly, Brilhante et al. [5] show a bottom-up approach to ontology design to support economic sustainability indicators.

Our work builds upon both examples. While we adopt, as orienting principles, two of the requirements listed by Kumazawa et al. [16] – interpretability and reusability – our ontology is designed to reflect as much as possible extant sustainability indicators, rather than our conceptualisation of the sustainability domain. In that respect, it follows the 'bottom-up' approach of Brilhante et al. [5]. However the overall goal of OSIS is to support *multiple* indicator systems as easily as possible. Hence our emphasis is upon the conceptual apparatus of such systems themselves, rather than the sustainability domain they represent. This results in OSIS having concepts such as "Indicator". To support this goal, we also use existing ontology engineering and evaluation approaches.

### 2.1 Ontology Engineering

Ontology engineering refers to a set of principles that are related to the development of an ontology for a specific domain. Janssen et al. [15] introduce an ontology engineering process that consists of *setup*, *design*, *approval* and *dissemination* phases. Various case studies have used aforementioned phases for their particular application. In 2000, the Gene Ontology (GO) Consortium [1] was established to produce a dynamic and controlled vocabulary in the context of the biomedical data that can be applied as knowledge of gene and protein roles. The GO[4] consists of three independent ontologies. Ryu and Choi [23] present an approach for term and taxonomy extraction based on information theory. Ferfeleder et al. [10] also introduce a semantic guidance to facilitate requirement categorisation process which is a challenging problem in the field of requirement engineering. Their semantic system provides a list of suggestions for engineers to define requirements effectively by using concepts, relations and axioms of a domain ontology.

Among various ontology engineering approaches, METHONTOLOGY is chosen for OSIS development due to its high rate of adoption among other domain ontologies. This approach is presented by Gómez-Pérez and her colleagues in the context of developing an ontology in the domain of chemicals [8]. METHONTOLOGY's framework enables the ontology engineers to construct the ontology at the knowledge level, and the design lifecycle is based on evolving prototypes consisting of three distinctive activities − namely **Management**, **Development** and **Support** − activities, which are performed either serially or in parallel. Management activities, such as *control* and *quality assurance*, are conducted at the start of the ontology development to identify the tasks to be performed, and the time and the resources required for their development activities shape the basis of the ontology. Ideally these phases − *specification, conceptualisation, formalisation, implementation* and *maintenance*− are performed through small incremental and iterative cycles. Finally, support activities which are carried out simultaneously with the development activities include: *knowledge acquisition, evaluation, documentation* and *configuration* [8]. We focus only upon the initial design and development of OSIS candidates − discussed in detail in Sections 3 and 4 − while some phases such as management, maintenance and documentation, although vital to the general creation of ontologies, are not discussed here.

### 2.2 Ontology Evaluation

Several approaches exist to evaluate ontologies, including: *Gold standard*, *Criteria-based* and *Task-based*. The first approach compares an ontology with a benchmark ontology. Maedche and Staab [18] propose a gold standard approach to empirically measure similarities between ontologies from different views such as lexical and conceptual aspects. Criteria-based approach evaluates the ontology based on the specific criteria such as *consistency*, *completeness*, *conciseness*, *expandability* and *sensitivity* [11]. Additionally, different researchers have proposed various ontology evaluation criteria [11, 12, 13]. The task-based approach evaluates an ontology based on the competency of the ontology in completing tasks, for example, whether the measured performance of that ontology within a specific application yields out the suitability of the ontology for that application.

One example of task-based ontologies is ROMEO (Requirements Oriented Methodology for Evaluating Ontologies) [25] which focuses on requirements as tasks. ROMEO is a compatible ontology evaluation technique which links generic requirements, such as "competency", "capability", "functionality" and "standardized" to evaluation measures through the development of some criteria "questions". This method is discussed in detail in Section 4.

## 3. DESIGNING WITH METHONTOLOGY

Here we step through the activities specified by METHONTOLOGY to support the development of two candidate ontologies for sustainability indicators. For clarity, we separate these activities into two processes, relating to the Pre-Design and Design phases of ontology development respectively.

[4]http://www.geneontology.org

| Domain (Subject) | Property (Predicate) | Rang (Object) |
|---|---|---|
| Superclass: Indicator+ | dc:title<br>dc:type<br>dc:description<br>dc:periodOfTime<br>dc:publisher<br>osis:instance-of<br>osis:hasCategory<br>osis:hasReference<br>osis:hasUnitofMeasurement<br>osis:hasIssue<br>osis:belongToIndicatorSystem | String<br>String<br>Superclass: Description<br>String<br>Superclass: IndicatorSet<br>Superclass: Indicator<br>Superclass: Category<br>Superclass: Reference<br>String<br>SuperClass:Issue<br>SuperClass:IndicatorSet |
| SuperClass: IndicatorSet+ | dc:title<br>dc:text<br>osis:hasIndicator | String<br>String<br>Superclass: Indicator |
| Subclass: Issue | dc:title<br>osis:isMeasuredByIndicator | String<br>Superclass: Indicator |
| SuperClass: Description+ | dc:title<br>dc:text<br>dc:format<br>dc:date<br>dc:publisher | String<br>String<br>String<br>Date<br>Superclass: IndicatorSet |
| SuperClass: Category+ | dc:title<br>dc:text<br>osis:hasIndicator | String<br>String<br>Superclass: Indicator |
| SuperClass: Reference+ | dc:title<br>dc:text<br>dc:isReferencedBy | String<br>String<br>Superclass: IndicatorSet |
| * : Abstract Class<br>+ : Concrete Class | | |

**Table 1: OSIS-Design A – Generic Model**

| Domain (Subject) | Property (Predicate) | Rang (Object) |
|---|---|---|
| Superclass: Indicator* | dc:title<br>dc:type<br>dc:description<br>dc:creator<br>dc:periodOfTime<br>osis:instance-Of<br>osis:hasCategory<br>osis:hasReference<br>osis:hasUnitofMeasurement | String<br>String<br>Superclass: Description<br>Superclass: Author<br>String<br>superclass: Indicator<br>Superclass: Category<br>Superclass: Reference<br>String |
| Subclass: GRI_Indicator+ | gri:hasCompilation<br>gri:hasDefinition<br>gri:hasDocumentation<br>gri:hasRelevance<br>gri:hasAspect | Subclass:GRI_Description<br>Subclass:GRI_Description<br>Subclass:GRI_Description<br>Subclass:GRI_Description<br>Subclass:GRI_Aspect |
| SubClass: OECD_Indicator+ | oecd:hasDefinition<br>oecd:hasInformation<br>oecd:hasTheme | Subclass: OECD_Description<br>Subclass: OECD_Description<br>Subclass:OECD_Theme |
| SuperClass: Description* | dc:title<br>dc:text<br>dc:format<br>dc:date<br>osis:instance-Of | String<br>String<br>String<br>Date<br>superclass: Description |
| SubClass: GRI_Description+ | gri:hasIndicator | Superclass: Indicator |
| SubClass: OECD_Description+ | oecd:hasIndicator | Superclass: Indicator |
| SuperClass: Category* | dc:title<br>dc:text<br>osis:instance-Of<br>osis:hasIndicator | String<br>String<br>superclass: Category<br>Superclass: Indicator |
| SubClass: GRI_Aspect+ | gri:hasIndicator | Superclass: Indicator |
| SubClass: OECD_Theme+ | oecd:hasIndicator | Superclass: Indicator |
| SuperClass: Reference* | dc:title<br>dc:text<br>dc:isReferencedBy<br>osis:instance-Of | String<br>String<br>Superclass: SustainabilitySet<br>superclass: Reference |
| * : Abstract Class<br>+ : Concrete Class | | |

**Table 2: OSIS-Design B – Specific Model**

## 3.1 OSIS Pre-Design Process

While METHONTOLOGYdoes not mandate a specific order, prior to beginning the formal design of OSIS we undertook *Specification* and *Knowledgement Acquisition* activities – as discussed below – which these provide key inputs into that design.

### 3.1.1 Specification

Gómez-Pérez et al. [8] state the purpose of the Specification phase is to produce an Ontology Requirements Specification Document (ORSD) in natural language, using informal, semi-formal or formal representation of the ontology. The ORSD identifies the scope, purpose and requirements of the ontology. It also specifies the level of formality required for the ontology, depending on whether terms and their meanings of the specific domain need to be codified in natural or formal language. We developed a specification of the OSIS ontology with the following purpose:

- **Purpose:** The aim of the ontology is to represent knowledge about sustainability indicators in the context of a specific application. The ontology can be reused for reasoning, reapplying and querying indicators for integration purposes.

### 3.1.2 Knowledge Acquisition

To translate the specification into design, we also conducted a Knowledge Acquisition stage, which METHONTOLOGY [8] emphasises as pivotal to developing a suitable domain ontology. Firstly, we consulted with sustainability experts associated with our broader project, through a series of interviews and workshops. Secondly, we analysed a number of available sources of domain knowledge, including widely used indicator sets, to extract key domain concepts to include in the ontology. Two indicator sets, the GRI and the OECD, were used to inform the initial design. The third system, taken from the UN, we selected as a frame of reference to evaluate OSIS.

Each system reflects subtle yet distinctive features of how sustainability is conceptualised by their respective organisations. Hence using 2+1 frames of reference allows us to triangulate the key domain concepts to at least a first degree of approximation. From this activity we identified the following key concepts for sustainability indicators including: *Core and Additional Indicators, Category, Description, Reference, Sustainability Set, Issue, Target, Objective, unitOfMeasurement, Title, ID, Organisation.* These key terms form the basis of the OSIS design we outline next.

## 3.2 OSIS Design Process

Having specified requirements and developed a set of key terms, we then began to design OSIS. Here we focus on three main phases suggested by the METHONTOLOGY approach [8] discussed below. These stages move us from a generic and abstract model of the domain through to progressively more specific design. During conceptualisation, we decided upon the need to develop two distinct conceptual models, each reflecting one of the two main requirements identified during the Specification phase. These two models are then formalised and implemented in the next two phases, which result in the candidate ontology designs shown in Table 1 and Table 2.

### 3.2.1 Conceptualisation

Conceptualisation is the most important step in ontology design [8]. The outcome of this phase is a specification of

the ontology components, including key *concepts, relations and instances*, which should reflect a set of terms produced in knowledge acquisition phase (see Section 3.1.2). A key task here is determining how specific indicator system data, taken from GRI and OECD, should be specified in relation to abstract concepts of "Organisation", "Indicator" and so on. Such relations ideally should reflect the requirements of the final ontology design.

We begin by developing a *taxonomy* of concepts, that is a hierarchical structure, representing concepts and the appropriate relationships between these concepts. A concept is an entity with a key role that can be inherited by other entities. In the taxonomy, we make no distinction between the ontological status of conceptual entities – whether for instance they ought to be represented by classes or individuals, or related by generic properties or specific sub-class relations.

We then represent these concepts as *classes* in the ontology. Following the taxonomic organisation, such concepts are distinguished between those which are broad and abstract (such as "Organisation", "Indicator"), which we model as superclasses of the ontology, and those which are specific and thus form the subclasses. Subclasses are inherited by superclasses, using *is-a* relationships. Additionally, classes may have properties that complement them. Where such properties express relationships between classes, such classes express a *has-a* relationship.

A key difficulty in translating such principles of ontology design is deciding whether relationships are of the *is-a* or *has-a* kind. Such decisions are influenced by different factors that also pertain to Object Oriented (OO) design in software engineering. In OO design, a class represents a set of *objects* − the terms object and instance are interchangeable − that share a common structure and a common *behavior* [3]. An object captures some well-defined behaviour from its class but it has a unique identity. The key point in designing an OO model is identifying the structure and behaviour of similar objects which construct the classes. Therefore, defining an entity as a class depends on whether its instances (objects) have common properties (behaviors).

In designing OSIS, this principle is reflected in the treatment of specific indicator classes, depending upon the emphasis. From the point of view of *reusability*, we see that system indicators can be included as instances of anonymous classes that extend "Indicator", and that are further specified by particular properties (e.g. "belongsToIndicatorSystem"). From the point of view of *explicitness*, we also see a usefulness in specifying particular system indicators as subclasses (e.g. "GRI_Indicator") of a generic indicator class (e.g. "Indicator"). The first view is more broad to cover sustainability indicators' key information with no reference to any particular organisations which is called ***Design A*** (shown in Table 1). The second view is more detailed to include direct references to specific indicator sets, which is called ***Design B*** (shown in Table 2). Both designs are discussed as follows:

1. ***Design A:***
   In this design, we sought to define broadly a suitable conceptual structure which reflects the generic key information of sustainability indicators. To a large extent, sustainability indicators are introduced to address issues of critical conditions in complex systems. In other words, indicators can provide solutions for such issues.
2. ***Design B:***
   In this design, our emphasis is on the organisations that develop sustainability indicators. Here we include as key conceptual constructs these organisations and their own indicator classifications. Therefore, a range of classes and relationships is specifically added for each sustainability indicator set.

### 3.2.2 Formalisation

This phase involves the transformation of the conceptual models defined previously into a formal representation as an ontology. This involves both the explicit representation of concepts and relations as classes, properties and individuals, and the development of *namespaces* for grouping related entities. Like *F-logic*, which is the basis of the ontology conceptualisation, namespaces are used to distinguish similar properties and relationships used in various ontologies from each other. The namespace declaration in ontology is similar to XML, where an alias is associated with the URI of a conceptual resource. We use the following namespaces in our designs:

- ***osis:*** refers to the URI[5] used to represent the most abstract and generic concepts and relations, such as <*osis:hasIndicator*> and <*osis:hasTarget*>.
- ***dc:*** refers to the Dublin Core metadata URI[6] to label common properties that pertain to most or all entities, for example <*dc:title*> for name entities and <*dc:type*> for type entities.
- ***gri:*** refers to the Global Reporting Initiative for the properties that are specifically related to GRI sustainability indicator sets, such as <*gri:hasAspect*> and <*gri:hasDocumentation*>.
- ***oecd:*** refers to the Organization for Economic Cooperation and Development for the properties that are specifically related to OECD sustainability indicator sets, for instance <*oecd:hasDefinition*> and <*oecd:hasTheme*>

The latter two namespaces are used only in Design B.

### 3.2.3 Implementation

To implement the conceptualisation, we first selected a suitable formal language to represent two OSIS design candidates. A knowledge representation language must have four essential features: "vocabulary", "syntax", "semantics" and "rules of inference" [17]. We have decided to represent the ontology in Resource Description Framework (RDF) [7] and Web Ontology Language (OWL) because of two reasons: 1) RDF/OWL ontologies are easily extensible by others and 2) RDF/OWL ontology data can be reasoned by computational agents and description logics using existing querying and visualisation tools, such as ontology editors, SPARQL libraries and third-party vocabularies.

**A) Technical Implementation:**
We also adopted particular technology tools to store and model the triples. PostgreSQL [8] was selected as a triple store

[5]http://www.cs.rmit.edu.au/knowledgebase/ontology/OSIS#
[6]http://purl.org/dc/elements/1.1/
[7]http://www.w3.org/RDF/
[8]http://www.postgresql.org/

framework, due to its compatibility with both relational (SQL) and semantic (RDF and OWL) languages. This simplified the loading of indicator sets, which were often expressed in loosely or unstructured forms, and therefore needed to be manipulated into formats such as CSV. We also used Protégé [9] to edit the two ontologies.

We then converted the two candidates into RDF/OWL form using Protégé. This involved converting each concept and relationship into equivalent semantic triple statements of *subject*, *object* and *predicate*. Each subject (e.g. a class or instance) is linked to an object (another class or instance) by a predicate (e.g. a "is-a" property). Subjects are considered as the *Domain* concepts of the property or relationship and objects are considered as the *Range*. Once specified, both ontologies were exported from Protégé into SDB2 and Postgres.

**B) Metadata Document Management:**
Once completed the technical implementation, we loaded both GRI and OECD indicator sets as instance data into both ontology candidates.

The GRI sustainability indicator set is presented in *eXtensible Business Reporting Language (XBRL)* [10]. XBRL is an XML-based language introduced to exchange business information. It uses the XML notation such as XML schema, XLink and XPath to express the semantic connections required in business reporting. The GRI organisation uses this language to define their sustainability metadata in a taxonomy that captures the individuals reporting concepts as well as the relationship between concepts and other semantic meanings in the original document. The details of our data transformation approach are given in Section 4.2.1.

# 4. EVALUATING WITH ROMEO

Having developed the two OSIS candidate ontologies and populated them with sustainability indicator data, we evaluated them with the four stages adapted from METHONTOLOGY.

## 4.1 OSIS Evaluation Process

In METHONTOLOGY, each ontology is evaluated with a collection of ontology frames of reference. Yu et al. [25] define a frame of reference $F$, $F =< F_c, F_i, F_r >$, where $F_c$ is the set of concepts, $F_i$ is a set of instances and $F_r$ is a set of relationships − which is the union between the set of relationships between concepts $F_{cr}$ and the set of relationships instances $F_{ir}$ − in a frame of reference. Here we interpret "frame of reference" to be the sorts of knowledge sources solicited during the Knowledge Acquisition activity – namely, the indicator systems themselves. Since the OSIS candidates are designed to support two key requirements of intuitiveness and reusability, accordingly we have chosen three indicator sets to evaluate the candidates against. The first two are the sources used to construct the candidates, the GRI and the OECD indicator systems. We refer to this below as the 'seen' frame of reference. The third one is one of the other sources, the UN indicator system. We refer to this as the 'unseen' frame of reference (since neither candidate has any explicit entities drawn from it). To conduct the evaluation, we designed several experiments to test both candidates against the high-level requirements described in Section 3.1.1, and further application and end-user requirements elicited through discussion with project stakeholders.

### 4.1.1 Establishing the Ontology Role

According to Yu et al. [25], eliciting the roles of an ontology is important to understand how the ontology is used in the context of an application and it also helps to determine a set of appropriate ontology requirements. The role of OSIS is defined as follows:

- **Ontology Role:** Enhance effectiveness of query expansion module in suggesting indicators for query tasks.

### 4.1.2 Ontology Requirements

Ontology requirements reflect a specific competency or quality of the ontology that can be obtained from existing ontology requirements or application requirements. In the context of sustainability indicators, we define two ontology requirements based on the ontology role and purpose.

- **Ontology Requirement 1:** Does the ontology provide a precise and intuitive representation of the indicator systems it represents?
- **Ontology Requirement 2:** Does the ontology allow for other indicator systems to be easily incorporated using existing concepts, properties and relations?

### 4.1.3 Criteria Questions

The ROMEO approach stipulates that a set of questions is administered for each of the requirements. Such questions explore various aspects of a given requirement providing a deeper understanding of the ontology. In addition, criteria questions lead to appropriate measures which are critical in an ontology evaluation context. Yu et al. [25] propose a list of criteria questions for a variety of ontology requirements. We specify two questions for the OSIS ontology candidates, and ensure each question is answered with respect to both seen (GRI and OECD) and unseen (UN) frames of reference. The criteria questions are listed below:

1. Do the ontology components (concepts, instances and relationships) adequately cover the terms of the given domain?
2. Do the ontology components (concepts, instances and relationships) capture the terms of the given domain correctly?

The first question examines the *coverage* criteria and the second question determines the *correctness* feature of the ontology.

### 4.1.4 Measures

At the final stage, Yu et al. [25] suggest adopting a set of measures that are compatible with the ontology requirements which allow us to answer the criteria questions. The ontology evaluation literature has proposed various ontology criteria and measure that are summarised in a previous survey conducted by Brank et al. [4]. Of these, we adopt the *precision* measure presented by Guarino [14] to measure the correctness criterion, by determining the percentage of overlapping terms in an ontology $O$ that overlaps with the set of terms from a frame of reference $F$ (Equation 1). Additionally, *recall* [14] is used to measure the coverage criterion,

[9] http://protege.stanford.edu/
[10] http://www.fxasustainability.com.au/

| | |
|---|---|
| **Example of a GRI indicator** | EN2 **Percentage of materials used that are recycled input materials.**<br>1. Relevance This Indicator seeks to identify the reporting organization's ability to use recycled input materials...<br>2. Compilation 2.1 Identify the total weight or volume of materials used as reported under EN1...<br>3. Definitions Recycled input materials: Materials that replace virgin materials that are purchased or obtained from internal or external sources...<br>4. Documentation Potential information sources include billing and accounting systems, the procurement or supply management department...<br>5. References OECD Working Group on Waste Prevention and Recycling. |
| **Example of XBRL for the above indicator** | <**label xlink:type**="resource" **xlink:label**="gri-core_EN02_lbl_en_terseLabel" **xlink:role**="http://www.xbrl.org/2003/role/terseLabel"<br>**xml:lang**="en" **id**="gri-core_EN02_lbl_en_terseLabel">EN2</**label**> <**label xlink:type**="resource" **xlink:label** ="gri-core_EN02_lbl_en_label"<br>**xlink:role**="http://www.xbrl.org/2003/role/label" **xml:lang**="en" **id**="gri-core_EN02_lbl_en_label">EN2</**label**><br><**label xlink:type**="resource" **xlink:label**="gri-core_EN02_lbl_en_guidelineDefinition"<br>**xlink:role**="http://www.globalreporting.org/2006/G3/guidelineDefinition"<br>**xml:lang**="en" **id**="gri-core_EN02_lbl_en_guidelineDefinition"> Percentage of materials used that are recycled input materials. (Core)</**label**><br><**label xlink:type**="resource" **xlink:label**="gri-core_EN02_lbl_en_protocolRelevance"<br>**xlink:role**="http://www.globalreporting.org/2006/G3/protocolRelevance"<br>**xml:lang**="en" **id**="gri-core_EN02_lbl_en_protocolRelevance"> This Indicator seeks to identify...</**label**><br>< **label xlink:type**= "resource" **xlink:label**="gri-core_EN02_lbl_en_protocolCompilation"<br>**xlink:role**="http://www.globalreporting.org/2006/G3/protocolCompilation"<br>**xml:lang**="en" **id**="gri-core_EN02_lbl_en_protocolCompilation"> 2.1 Identify the total weight or volume of materials used as reported under EN1.<br>2.2 Identify the total weight or volume of recycled input materials. If estimation is required ...</**label**><br>... |
| **RDF triples after applying SAX on XBRL file** | <EN2> <http://www.w3.org/1999/02/22-rdf-syntax-ns#**type**> <GRI_Indicator>.<br><EN2> <http://pur1.org/dc/elements/1.1/**description**> "Environment".<br><EN2> <http://pur1.org/dc/elements/1.1/**format**> "pdf".<br><EN2> <http://pur1.org/dc/elements/1.1/**isReferencedBy**> <EN2-Reference>.<br><EN2> <http://pur1.org/dc/elements/1.1/**title**> "Percentage of materials used that are recycled input materials".<br><EN2> <http://www.cs.rmit.edu.au/knowledgebase/ontology/OSIS/**hasUnitOfMeasurement**> "percentage".<br><EN2> <https://www.g1oba1roporting.org/**hasAspect**> <EN2-Aspect>.<br><EN2> <https://www.g1oba1roporting.org/**hasRelevance**> <EN2-Relevance>.<br><EN2> <https://www.g1oba1roporting.org/**hasCompilation**> <EN2-Compilation>.<br>...<br><EN2-Relevance> <http://pur1.org/dc/elements/1.1/**text**> "This Indicator seeks to identify the reporting organization's ability to use recycled input materials...".<br><EN2-Compilation> <http://pur1.org/dc/elements/1.1/**text**> "2.1 Identify the total weight or volume of materials used as reported under EN1...".<br>... |

**Table 3: An example of a GRI indicator and a snapshot of its XBRL and RFD representation**

referring to the percentage of overlap between a set of terms from the ontology and the frame of reference (Equation 2).

$$precision(O, F) = \frac{|F \cap O|}{|O|} \quad (1)$$

$$recall(O, F) = \frac{|F \cap O|}{|F|} \quad (2)$$

These metrics are originally given from Information Retrieval literature − known as *retrieval performance evaluation* [2] − to determine the quality of the answer set generated from a query task. However they are also applied in ontology context. For example, Rodriguez and Egenhofer [22] use these metrics to measure the fraction of similar entity classes from different ontologies, and Euzenat [9] also uses such metrics for measuring ontology alignments.

The $F$-measure is the harmonic mean of precision and recall that provides a sense of adequacy and balance modelling of the domain being presented in the ontology. Yu et al. [25] apply the $F$-measure to ontology evaluation in the context of indicating appropriate coverage of concepts in the relevant frame of reference. They give Equation 3 as follows:

$$F\text{-}measure(O,F) = \frac{2}{\frac{1}{recall(O,F)} + \frac{1}{precision(O,F)}} \quad (3)$$

## 4.2 Experiments

In evaluating the OSIS, we perform six sets of experiments. We select indicators from the category of *Economy* from three frames of reference (the GRI, the OECD and the UN indicator sets). We then compare $F$-measures for both OSIS ontology candidates, Design A and Design B, against the frames of reference.

### 4.2.1 *Preparing and Analysing Documents*

To show these steps in practice, we include an example of a GRI indicator from the Environment category and its XBRL and RDF representation in Table 3. As described in section 3.2.3, the GRI sustainability indicator set is represented in XBRL. First, we phrased the required data from this document using SAX functions including: *start-document, start-element, end-element, character and end-document* by applying on *label* tags with conditioning the relevant attributes such as *id*, *xlink* : *type* and *xlink* : *label*. Second, the ontologies are populated with indicator data with the use of Protégé interface, for instance some of the ontological properties as described in Table 2 and shown in bold font in Table 3 are *description, format, isReferencedBy*. Next, we export the RDF file − the final ontology document − from Protégé. An example of the generated file in *NTriples*[11] is shown in Table 3 that describes the triple statements consisting of subject, predicate and object which represents the key information of $EN2$ using relevant namespaces, concepts and appropriate relationships in Design B.

The final step of the experiment obtains the overlapping terms between the original frames of reference and the ontology documents. In order to produce consistent results, a pre-processing stopping algorithm and Porter's Stemmer technique [20] are applied to the resulting overlapping terms. We also use the algorithm presented by Broder [6], which determines the syntactic similarity between two documents. In our case, the first document is the textual representation of the ontology, and the second is the frame of reference. Each document is considered as a sequence of tokens, divided into the number of contiguous subsequences called *shingles*, of length $n$ that is also known as *n-gram*. The algorithm compares the sets of *n-grams* from two documents and calculates their resemblance value.

[11] http://www.w3.org/2001/sw/RDFCore/ntriples/

| Frame of Reference | $\|F\|$ | OSIS Ontology | $\|O\|$ | $\|F\| \cap \|O\|$ | $Precision_{ave}$ | $Recall_{ave}$ | $F\text{-}Measure$ |
|---|---|---|---|---|---|---|---|
| $GRI-Frame$ | 2560 | $Design-A$ | 2280 | 1602 | 0.702 | 0.625 | **0.661** |
| | 2560 | $Design-B$ | 2309 | 2090 | 0.905 | 0.816 | **0.854** |
| $OECD-Frame$ | 986 | $Design-A$ | 802 | 590 | 0.735 | 0.598 | **0.659** |
| | 986 | $Design-B$ | 890 | 765 | 0.859 | 0.877 | **0.867** |
| $UN-Frame$ | 500 | $Design-A$ | 445 | 325 | 0.650 | 0.733 | **0.682** |
| | 500 | $Design-B$ | 303 | 247 | 0.494 | 0.315 | **0.307** |

**Table 4: Results for OSIS, GRI Frame, UN Frame, precision, recall and *F*-measure**

# 5. RESULTS

In Table 4, we present results from experiments described in the previous section. We use the average $F$-measure along with associated metrics, the average recall and precision, to compare the two OSIS design candidates against seen (GRI and OECD used in Design B) and unseen (UN not used in Designs A and B) frames of reference.

The number of terms ($|F|$) between the three frames is different due to a number of reasons. For example, in a comparison with the GRI and the UN, while both frames distinguish between economic, environmental and social indicators, the GRI is directed largely towards corporate sustainability reporting, while the UN indicator set is aimed at measuring nation-level sustainability development. The GRI therefore includes more economic indicators, while the UN emphasises social indicators. The UN set also includes a fourth category of 'institutional' indicators, which inflates the overall indicator count. This consequently affects the number of terms in each ontology $|O|$ and overlapping terms $|F| \cap |O|$ for each set of experiments.

The graph in Figure 1 features the results for coverage using the $F$-measure. Comparing the results for the GRI and the OECD with the UN frames reveals similar coverage (approx. 65%) for Design A. By contrast, the $F$-measure shows significant difference between the two frames for Design B; the GRI-Frame and OECD-Frame have large proportions of coverage (on average 85%) whereas the UN-Frame's number declines significantly (30%).

These figures can be explained by comparing the design decisions for the two candidates. Design A presents a generic model for the ontology with no direct reference to any sustainability indicators. Actual indicators are assumed to instantiate, rather than inherit, from the *Indicator* class. We view this accordingly as a more terse and generic conceptualisation of the domain. By contrast, Design B presents a specific model, with more class references to particular sustainability indicators. This results in a higher $F$-measure against the seen frames of reference (GRI and OECD ) – but because the specific wording of concepts maps directly to that frame, it performs more poorly against the unseen frame of reference (UN).

The contrasting results map intuitively to different requirements that can be said to underwrite the construction of the two ontology candidates. Where an ontology needs to be *precise* and *transparent* – to faithfully represent, at a conceptual level, the specific conceptualisation of given indicator system – the approach adopted by Design B is preferred, since it results in better $F$-measures where that system is used as a frame of reference. In contrast, Design A better supports cases where the requirements emphasise ontology *reuse* with minimal cost of extension or refactoring, since it performs better against unseen frames of reference. We also acknowledge there are cases where a compromise between these requirements might result in a hybrid of Designs A and B. Indeed, such an option (maximising $F$-measures against both seen and unseen frames of reference) might be preferred where cost and time constraints permit.

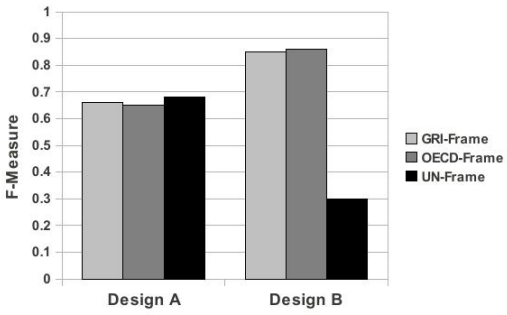



**Figure 1: Results showing granularity using *F*-measure, GRI and OECD are used in constructing the ontology and UN is not used.**

# 6. CONCLUSION AND FUTURE WORK

In this paper, we briefly introduce the field of sustainability indicator systems, and argue that ontologies are well-suited for representing such systems formally. We adopt METHONTOLOGY, a well-known ontology engineering methodology, to guide the development of two pilot ontology candidates for this domain. We then apply ROMEO to evaluate the candidates. The evaluation consisted of precision and recall to test the degree of coverage of indicators against two frames of reference. These metrics are also used in other ontology evaluation research [9, 14, 22, 25]. The first two frames of reference (GRI and OECD) were used to construct Design B; the third (UN) was only used in the evaluation.

The two candidates, A and B, differ largely in terms of abstraction. Design A applies an object-oriented style approach. Here, for example, the concept *indicator* is defined as a class, while specific instances of indicators are treated as individuals which instantiate properties and relationships of the *Indicator* class. By contrast, Design B treats each indicator instance as a class as well. Accordingly, they inherit rather than instantiate properties and relationships of the *Indicator* class. This produces a much larger ontology that maps directly to the specific frames of reference that it is derived from. Accordingly, Design B scores higher $F$-measure results against the *seen* frames of reference (the GRI and the OECD have been used in constructing this model). However, as our results show, Design A produces a better $F$-measure score against an *unseen* frame of refer-

ence, such as the UN that has not been used in informing this model.

We conclude that Design B is preferable where the domain requirements require a high degree of fidelity to seen frames of reference, while Design A offers greater reuse in contexts where unseen sets of indicators need to be added to the ontology in an ad hoc fashion. As sustainability indicators themselves continue to evolve, for this specific domain we argue Design A is preferable; though as consensus builds among reporting organisations, we also anticipate the possibility of blending both approaches in future. More generally, we show that both METHONTOLOGY and ROMEO can be productively used to guide the design and evaluation of domain-level ontologies, and that quantitative measures such as $F$-measures can be used to develop heuristics for preferring one ontology candidate to another, given a set of requirements and frames of reference.

We anticipate further work along several lines. Firstly, we think that ROMEO can be linked to METHONTOLOGY in a more systematic way, to guide ontology development from requirements through to evaluation and selection. Secondly, ROMEO itself can be extended through the sorts of quantitative procedures we apply here. In addition, a user-study is required to further evaluate the two design candidates with the real-world scenarios. In order to receive feedback from expert and non-expert users, one scenario can be applying Designs A and B on a knowledge base of a web-based application that reasons and queries indicators for various purposes. Finally, with reference to the field itself, further work can be undertaken to incorporate additional sustainability indicators systems, and to further refine the candidate OSIS ontologies presented here.

# 7. REFERENCES


[1] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, et al. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, 25(1):25–29, 2000.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[3] G. Booch. *Object-oriented analysis and design with applications*. Benjamin-Cummings, 1994.

[4] J. Brank, M. Grobelnik, and D. Mladenic. A survey of ontology evaluation techniques. In *Conference on Data Mining and Data Warehouses (SiKDD)*, 2005.

[5] V. Brilhante, A. Ferreira, J. Marinho, and J. Pereira. Information integration through ontology and metadata for sustainability analysis. In *The International Environmental Modelling and Software Society (iEMSs) 3rd Biennial Meeting*, 2006.

[6] A. Broder. Identifying and filtering near-duplicate documents. In *Combinatorial Pattern Matching*, volume 1848 of *LNCS*, pages 1–10. Springer, 2000.

[7] B. Chandrasekaran and T. R. Johnson. Generic tasks and task structures: history, critique and new directions. In *Second generation expert systems*, pages 232–272. Springer, 1993.

[8] O. Corcho, M. Fernández-López, and A. Gómez-Pérez. Methodologies, tools and languages for building ontologies.where is their meeting point? *Data and Knowledge Engineering*, 46(1):41–64, 2003.

[9] J. Euzenat. Semantic precision and recall for ontology alignment evaluation. In *Proc. 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 348–353, 2007.

[10] S. Farfeleder, T. Moser, A. Krall, T. Stålhane, I. Omoronyia, and H. Zojer. Ontology-driven guidance for requirements elicitation. In *The Semanic Web: Research and Applications*, volume 6644 of *LNCS*, pages 212–226. Springer, 2011.

[11] A. Gómez-Pérez. Towards a framework to verify knowledge sharing technology. *Expert Systems With Applications*, 4(8):519–529, 2007.

[12] T. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[13] N. Guarino. Some ontological principles for designing upper level lexical resources. In *1st International Conference on Language Resources and Evaluation*, pages 527–534, 1998.

[14] N. Guarino. Towards a formal evaluation of ontology quality. *IEEE Intelligent Systems*, 19(4):74–81, 2004.

[15] S. Janssen, F. Ewert, and L. Hongtao. Defining assessment projects and scenarios for policy support: Use of ontology in integrated assessment and modelling, environmental modelling and software. *Special issue on simulation and modelling in the Asia-Pacific region (ASIMMOD)*, 24:1491–1500, 2009.

[16] T. Kumazawa, O. Saito, K. Kozaki, T. Matsui, and R. Mizoguchi. Toward knowledge structuring of sustainability science based on ontology engineering. *Sustainability Science*, 4(1):99–116, 2009.

[17] H. J. Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23(2):155–212, 1984.

[18] A. Maedche and S. Staab. Measuring similarity between ontologies. In *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, volume 2473, pages 15–21. Springer, 2002.

[19] D. Maynard, W. Peters, and Y. Li. Metrics for evaluation of ontology-based information extraction. In *Workshop on Evaluation of Ontologies for the Web(EON), Scotland*, 2006.

[20] M. F. Porter. An algorithm for suffix stripping. In *Readings in information retrieval*, pages 313–316. Morgan Kaufmann Publishers Inc., 1997.

[21] D. Richards. The reuse of knowledge: a user-centred approach. *International Journal of Human-Computer Studies*, 52(3):553–579, 2000.

[22] M. Rodriguez and M. Egenhofer. Determining semantic similarity among entity classes from different ontologies. *Knowledge and Data Engineering, IEEE Transactions on*, 15(2):442–456, 2003.

[23] P. Ryu and K. Choi. Taxonomy learning using term specificity and similarity. *Workshop on Ontology Learning and Population*, pages 41–48, 2006.

[24] L. Steels. The componential framework and its role in reusability. In *Second generation expert systems*, pages 273–298. Springer-Verlag, 1993.

[25] J. Yu, J. A. Thom, and A. Tam. Requirements-oriented methodology for evaluating ontologies. *Information Systems*, 34(8):766–791, 2009.

# Graph-based Concept Weighting for Medical Information Retrieval

Bevan Koopman[1,2], Guido Zuccon[1], Peter Bruza[2], Laurianne Sitbon[2], Michael Lawley[1]

[1]Australian e-Health Research Centre, CSIRO, Brisbane, Australia

[2]Faculty of Science & Technology, Queensland University of Technology, Brisbane, Australia

{b.koopman, p.bruza, laurianne.sitbon}@qut.edu.au, {guido.zuccon, michael.lawley}@csiro.au

## ABSTRACT

This paper presents a graph-based method to weight medical concepts in documents for the purposes of information retrieval. Medical concepts are extracted from free-text documents using a state-of-the-art technique that maps n-grams to concepts from the SNOMED CT medical ontology. In our graph-based concept representation, concepts are vertices in a graph built from a document, edges represent associations between concepts. This representation naturally captures dependencies between concepts, an important requirement for interpreting medical text, and a feature lacking in bag-of-words representations.

We apply existing graph-based *term* weighting methods to weight medical concepts. Using concepts rather than terms addresses vocabulary mismatch as well as encapsulates terms belonging to a single medical entity into a single concept. In addition, we further extend previous graph-based approaches by injecting domain knowledge that estimates the importance of a concept within the global medical domain.

Retrieval experiments on the TREC Medical Records collection show our method outperforms both term and concept baselines. More generally, this work provides a means of integrating background knowledge contained in medical ontologies into data-driven information retrieval approaches.

## Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.3 Information Search and Retrieval

## General Terms

Theory, Experimentation

## Keywords

Medical Information Retrieval, Graph Theory



*ADCS'12*, December 05 – 06 2012, Dunedin, New Zealand.



## 1. INTRODUCTION

Most information retrieval (IR) models represent documents as bag-of-words, that is, the representation does not consider word order or term dependence. However, alternative representations, such as graph-based representations have shown that taking term dependence into account can improve retrieval performance [3]. In these approaches a document is modelled as a graph, where terms are vertices and edges represent relations between terms. The importance of a term within a document is proportional to its connectedness to other terms and can be estimated with graph-based measures such as the PageRank algorithm [11].

At the same time there is an increasing body of research within the IR community focused on systems for medical information retrieval [6]. The nature of medical natural language presents some specific challenges — vocabulary mismatch is more prevalent and there is greater interdependence between terms (e.g., between diseases and treatments or organisms and diseases) [12, 7]. This motivates the use of alternative IR models that incorporate more semantic approaches to capture the innate dependencies between terms in medical natural language.

In this paper we apply existing graph-based term weighting approaches to medical IR. Rather than applying these approaches to the original term representation of documents, we first convert the documents into medical concepts defined by the SNOMED CT medical ontology. The motivation for this conversion is that concept-based representations have a proven track record in medical IR [19, 9, 7]. Concepts (the counterpart of terms in this context) are weighted according to their connectedness within the graph using an adapted PageRank algorithm. In addition, we propose a novel background weighting method that incorporates the importance of the concept within the global medical domain (rather than just a single corpus); this is done by injecting domain knowledge from the SNOMED CT ontology into the weighting function. A consequence of this method is that a large number of query concepts are actually excluded, which proves effective as a query concept selection method.

The remainder of the paper is organised as follows: Section 2 provides the background on graph-based IR and concept-based representations for medical IR. Section 3 details our graph-based concept-weighting model, including the injection of domain knowledge from the SNOMED CT ontology in the weighting function. Section 4 describes the evaluation methodology using the TREC Medical Record Track and presents results. Section 5 discusses our findings and considers future work.

## 2. BACKGROUND

This section provides (i) background and related work on concept-based representations of documents for medical IR; and (ii) graph-based term weighting methods for information retrieval. The following section will present our model which combines these two approaches and extensions by the injection of domain knowledge.

### 2.1 A 'Bag-of-Concepts' Model for Medical IR

Broadly, concept-based IR aims to make use of external knowledge sources (such as thesauri or ontologies) to provide additional background knowledge and context that may not be explicit in a document collection and users' queries. Performance in concept-based IR is highly dependent on the specific domain model or ontology used. General applications (those that utilise WordNet or Open Directory) struggle to outperform keyword-based systems [16, 13, 5]. However, biomedical applications — which use domain specific ontologies — do demonstrate consistent improvements [19, 9, 7]. Generally, concept-based approaches fall into two categories: (i) Those that maintain the original term representation of documents and only utilise concept-based representations (typically of the query) at retrieval time. The query expansion method of Liu et al. [9] is an example of this. (ii) Approaches that translate the original terms in a document into concepts prior to indexing. Zhou et al. [19], Egozi et al. [5] and Koopman et al. [7] take this approach, thereby utilising a 'bag-of-concepts' representation of a document; they demonstrate significant improvements over a term baseline. This latter approach is the one we adopt to develop a graph-based *concept* weighting model. Therefore, we provide some additional details of the 'bag-of-concepts' model below.

The conversion of text to concepts is achieved by a natural language processing system called MetaMap [1], developed by the U.S. National Library of Medicine. MetaMap analyses biomedical free-text and identifies concepts belonging to Unified Medical Language System (UMLS). MetaMap is widely adopted in clinical NLP [10] and IR [6, 9]. Using MetaMap, both queries and documents are converted, hence the 'bag-of-concepts' representation. For example, the text '`vascular dementia`' found in a document would be replaced with the UMLS concept id `C0011269`; Koopman et al. [7] provide further details of this process.

As with the 'bag-of-words' representation, the 'bag-of-concepts' does not incorporate the innate dependencies between concepts that exist in medical natural language. An alternative to bag-of-words representations are graph-based representations of documents, which aim to represent relations between terms in a document as edges in a document graph [3]. We now consider previous graph-based approaches with an eye for how they might be applied to our bag-of-concepts representation, thus capturing the innate relations that may exist between medical concepts.

### 2.2 Graph-based Term Weighting

Graph-based models have been applied in information retrieval, generally as part of connectionist approaches [4]. Shifting weights between vertices in a graph is the basis for the Inference Network model of Turtle & Croft [14], and the basis for the InQuery language used as part of the popular search engine Lemur[1]. Graphs provide a convenient means of representing information for IR applications — the propagated learning and search properties of a graph provide a powerful means of identifying relevant information items [3] (be they terms or documents). Graph-based algorithms, such as the popular PageRank algorithm [11] are examples of graph theoretic properties that can be utilised very effectively in a information retrieval scenario.

Blanco & Lioma [3] developed a graph-based term weighting model that represents each document as a graph: vertices are terms and edges are relations between terms. Relations may be simple co-occurrence relations within a context window, or more complex grammatical relations. The importance of a term within a document can then be estimated by the number of related terms and their importance, much in the same way PageRank estimates the importance of a page via the pages that link to it.

We hypothesise that Blanco's model adapted to a concept representation of documents may be a powerful tool for medical IR as it would capture the dependencies between concepts found in medical free-text. We therefore integrate Blanco & Lioma graph-based term-weighting model into previous concept-based approaches to medical IR, this is done in the next section. The remainder of this section provides an explanation of the original graph-based model and provides an example of its application on an exerpt of medical text.

In Blanco & Lioma's graph-based term weighting model, a term $i$ in a document is represented by the vertex $v_i$. A vertex is connected to other vertices, $\mathcal{V}(v_i)$ denoting the set of vertices connected to $v_i$. The weight of $v_i$ within a document is initially set to 1 and the following PageRank function is run for several iterations

$$S(v_i) = (1-\phi) + \phi * \sum_{v_j \in \mathcal{V}(v_i)} \frac{S(v_j)}{|\mathcal{V}(v_j)|} \quad (0 \leq \phi \leq 1) \qquad (1)$$

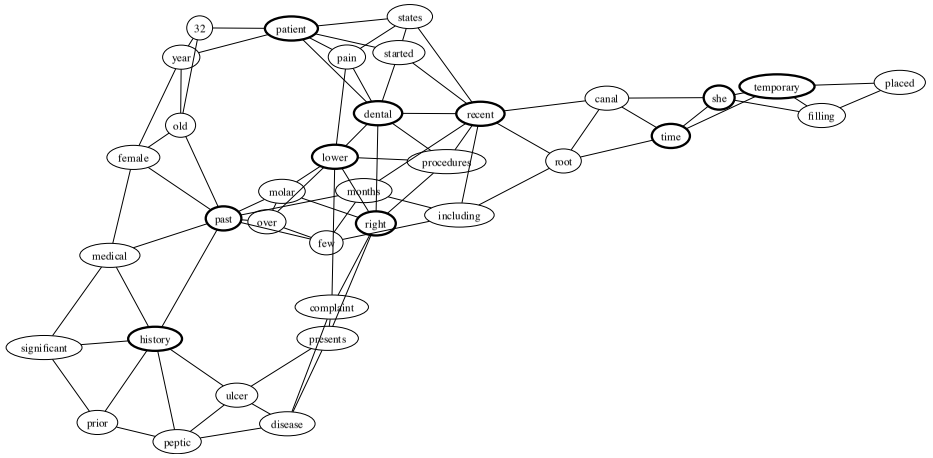
where $\phi$ is the damping factor which controls "vote recycling" from the original PageRank algorithm [11]. Blanco & Lioma showed that only a small number of interations ($< 50$) is required to obtain convergence [3]. Edges between vertices are based on relations between terms. Term relations can be implemented as the co-occurrence between two terms within a set context window $N$[2].

Next, we present an example of the graph produced when this method is applied to a small sample of medical text; this is done to highlight some of the characteristics of graph-based representations. Firstly, an example medical text document is shown in Figure 1(a). From this sample text Figure 1(b) shows the corresponding graph built using a context window of $N = 3$ terms in total. The vertex scoring algorithm of Equation 1 is applied to each vertex and the ten terms with the highest score are highlighted. These include the terms `dental`, `patient` and a number of temporal terms (`history`, `past`, `time`, `recent`). Those terms with higher scores provide an indication of the important terms appearing in this document. The next section shows how this information is included into the retrieval method.

[1]Lemur Project, `http://www.lemurproject.org`.
[2]Other relations may consider grammatical modifiers or part-of-speech information.

"The patient is a 32-year-old female with a past medical history significant for a prior history of peptic ulcer disease who presents with a complaint of right lower dental pain. The patient states that she was started on recent dental procedures, on a right lower molar, over the past few months, including a recent root canal, at which time she had a temporary filling placed."

(a) Example medical text document.




(b) Term graph of example medical text document; stop words removed.

**Figure 1: Resulting term graph 1(b) built from the above medical document 1(a). Built using co-occurrence window $N = 3$. Bolded nodes indicate the 10 terms with greatest score within the document (according to Equation 1).**

### 2.2.1 Retrieval Function

The graph-based vertex score of Equation 1 is now integrated into a retrieval function. Typical retrieval functions estimate the relevance between a document and a query as

$$R(d,q) \approx \sum_{t \in q} w(t,q) * w(t,d) \tag{2}$$

where $w(t,q)$ is the weight of the term in query, often uniform for ad-hoc queries, thus $w(t,q) = 1$. The second component, $w(t,d)$, is the weight of the term in the document. The graph-based score provides a means of estimating $w(t,d)$

$$w(t,d) = idf(t) * S(v_i) \tag{3}$$

where $S(v_i)$ is the vertex score from Equation 1 and $idf(t)$ is the inverse document frequency of the term. The retrieval function from Equation 2 can be reexpressed as

$$R(d,q) = \sum_{t \in q} w(t,d) \tag{4}$$

In the next section we apply the graph-based term weighing method to the use of concept-based representations and later show how doing so improves the performance of a medical IR system.

## 3. GRAPH-BASED CONCEPT WEIGHTING

Building a graph of concepts is done in the same way as building a graph of terms: a context window of fixed length is moved across a document, concepts which co-occur within the context window are connected with an edge in the graph of concepts. Although the process of creating the graph for terms and concepts is the same, the resulting graph itself can differ significantly for the concept representation. To demonstrate this we revisit the example text document and resulting graph from Figure 1. Converting the example text document to concepts and constructing the graph results in the graph illustrated in Figure 2. The concepts are identified by their concept id in both the document and the graph, but we also include their description in parentheses to make the example readable. The PageRank function from Equation 1 is applied and the 10 vertices with the highest scores are highlighted in the figure.

There are many more concepts in the concept graph than terms in the term graph. This is because a single term can map to multiple concepts, for example, the term *HIV* maps to three concepts: `C0019682` *HIV Virus*, `C0019693` *HIV (Disease)* and `C0019699` *HIV+ finding*. Alternatively, multiple terms can map to a single concept, for example, the phrase *Peptic ulcer disease* maps to the single concept `C0030920`.

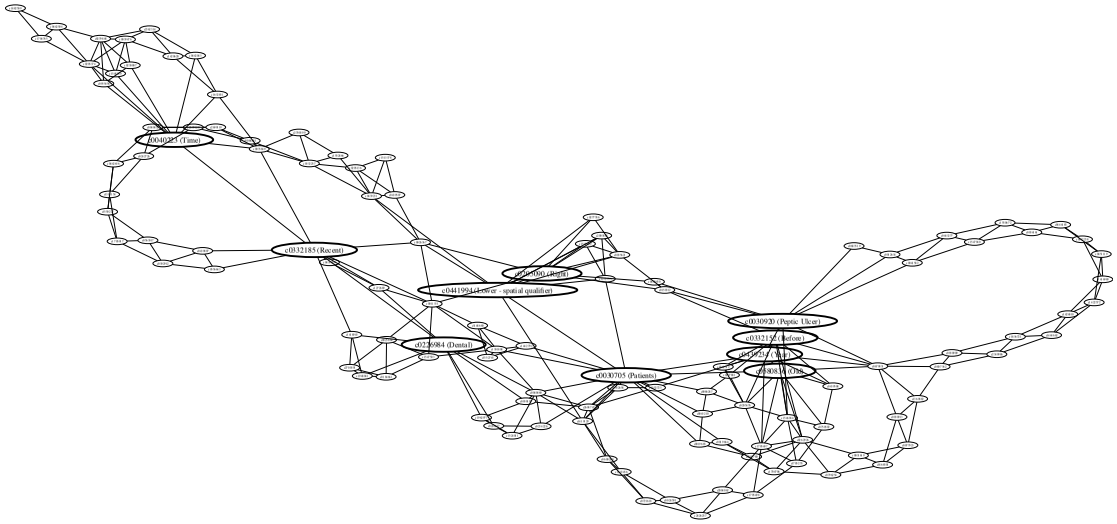



**Figure 2: Resulting concept graph built from the medical document from Figure 1(a). Built using co-occurrence window $N = 3$. Bolded nodes indicate the 10 concepts with greatest score within the document (according to Equation 1).**

Comparing the term graph from Figure 1 and the concept graph from Figure 2 we observe that both contain similar high score items — `dental` appears in both, as does `patient` and temporal items like `history`, `year`, `recent` and `time`. However the one major difference is the concept `Peptic Ulcer`, which appears in the concept graph, but not in the term graph. The reason for this is twofold: firstly, when converting to concepts the n-gram `peptic ulcer` from the original text maps to the single concept `c0030920`; secondly, when represented in graph form the concept is highly connected and therefore receives a high score. `Peptic Ulcer`'s high score reveals it as an important concept within the concept graph (and therefore this document), a feature not present in the term graph.

## 3.1 Concept Retrieval Function

Applying the weighting and retrieval functions to concepts we simply substitute terms for concepts. Thus, the original term weighting function from Equation 3 is updated to weight a concept $c$ within document $d_c$ as

$$w(c, d_c) = idf(c) * S(v_i) \tag{5}$$

The original retrieval function is updated to

$$R(d_c, q_c) = \sum_{c \in q_c} w(c, d_c) \tag{6}$$

where $d_c$ is the document converted to concepts and $q_c$ is query converted to concepts.

## 3.2 Injecting Domain Knowledge into the Weighting Function

The health informatics community has invested considerably in the development of medical domain knowledge resources, for example, the SNOMED CT ontology. These resources describe in great detail[3] the coverage of topics and terminology used within the medical domain. Incorporation of this large external resource into an IR system is not a trivial task. However, if effective integration can be achieved the IR system could potentially make far more informed judgements regarding relevance when presented with a user's query. Towards this goal, this section describes a method for injecting domain knowledge into the weighting function.

The concepts in our concept-based graph model are taken from the SNOMED CT medical ontology. SNOMED CT also defines explicit relationships between concepts, for example the *HIV* virus concept is related to the *AIDS* disease concept. SNOMED CT therefore can also be modelled as a graph, with concepts as vertices and relationships as edges. A concept's number of edges can be an indicator of the concept's importance within the medical domain. Consider the simple example for the concept *Asthma*, which is related to 50 different other concepts, a subset of which are shown in Figure 3.

Concepts important to the medical domain, such as diseases and treatments, are carefully modelled by the designers of SNOMED CT and contain detailed relationships to other concepts. In contrast, concepts that are peripheral to the medical domain are only broadly defined and typically contain only a small number of relationships. In contrast to the *Asthma* example, SNOMED CT defines the concept *Dog*,

[3]SNOMED CT contains approximately 311,000 concepts and 1,360,000 relationships between concepts.

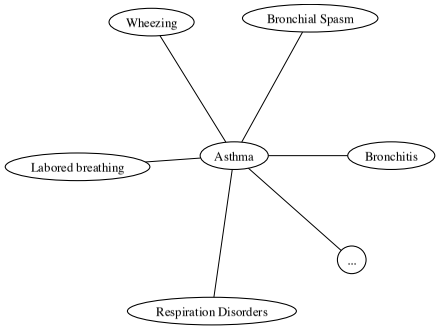



**Figure 3: The concept *Asthma* is related to 50 other concepts in the SNOMED CT ontology, an indication of its importance within the medical domain.**

which is related to only 5 other concepts — an indication it may be of lesser importance.

Identifying important concepts within the medical domain may provide an indication of what users may be interested in when searching medical documents. We would like to include this indication of importance within the medical domain into our graph-based concepts weighting model. Currently, the concept weighting scheme is based on the number of related concepts within the graph built for a single document. This method captures the importance of the concept within a document, but does not consider the importance of the concept within the wider medical domain. The original concept weight can be adjusted by the number of related concepts within the SNOMED CT ontology, representing its 'background' importance within the medical domain. The weighting function $w(c, d_c)$ of Equation 5 can then be augmented as

$$w(c, d_c) = idf(c) * S(v_i) * \log(|\mathcal{V}_s(c)|) \quad (7)$$

where $\mathcal{V}_s(c)$ is the set of edges adjacent to concept $c$ in the SNOMED CT ontology graph. A concept's weight is therefore adjusted based on its background weight within the medical domain, similar to the way background smoothing is applied in language models based on a term's frequency within the corpus. However, the weighting using SNOMED CT is independent of the document corpus and utilises a global measure of importance for the concept within the medical domain.

# 4. EMPIRICAL EVALUATION

This section details our experimental setup and evaluation methodology; results are presented in the next section.

## 4.1 Test Collection

As the test collection we use the TREC 2011 Medical Records Track, a collection of 100,866 clinical record documents from U.S. hospitals. Documents belonging to a single patient's admission were treated as sub-documents and were concatenated together into a single document called a patient *visit* document. This was done because the unit of retrieval in TREC 2011 MedTrack was a patient visit rather than individual report. Collapsing reports to patient visits was a common practise among many TREC MedTrack participants [17]. The corpus then contained 17,198 patient visit documents.

| Corpus | #Docs | Avg. doc. len. | #Vocab. |
|---|---|---|---|
| MedTrack: | | | |
| Terms | 17,198* | 2338 terms/doc | 218,574 |
| Concepts | 17,198* | 6066 concepts/doc | 54,143 |

*100,866 original reports collapsed to 17,198 patient *visit* documents.

**Table 1: Collection statistics for the TREC 2011 MedTrack corpus of clinical patient records. Statistics are provided for the original term corpus and subsequent corpus after conversion to concepts using the information extraction tool MetaMap.**

The original textual documents were translated into concept identifiers using the information extraction system MetaMap, as outlined in Section 2.1[4]. Statistics for both the term and concept corpora are provided in Table 1.

## 4.2 Baselines for Comparison

We implement a number of baselines for comparison against our graph-based concept weighting model:

**terms-tfidf:** We consider a state-of-the-art bag-of-words model. In initial experiments a tf-idf implementation actually demonstrated the best performance over BM25 and a Language Model with Dirichlet smoothing. Thus, we adopt as a baseline the Lemur variant implementation of tf-idf (which uses the Okapi TF formula [18]; parameterising document length normalisation with $b$ and term frequency weighting with $k1$). This baseline was tuned by selecting the best performing (oracle) pair of parameters values for $b$ and $k1$ from a complete sweep of the parameter space in the ranges $b = [0, \ldots, 1]$ (with increments of 0.1) and $k1 = [0, \ldots, 40]$ (with increments of 1). The best values were $b = 0.45$ and $k1 = 3.7$. This strong tf-idf tuned baseline is denoted terms-tfidf.

**terms-graph:** We implemented Blanco & Lioma's graph-based weighting method and apply it to terms. The damping factor parameter $\phi$ from Equation 1 is set to 0.85 according to the findings of Blanco & Lioma [3]. Similarly, the number of iterations and the context window size were set at 20 and 10 respectively, in line with Blanco & Lioma. This baseline is denoted terms-graph.

**concepts-tfidf:** We implement a bag-of-concepts model as the same tf-idf model as for terms-tfidf, but on the concepts corpus (as opposed to the term corpus). Parameters for this baseline were tuned in the same manner as terms-tfidf; $b = 0.35$, $k1 = 5.0$. This tuned baseline is denoted concepts-tfidf.

[4]Koopman et al. found that mapping to the SNOMED CT subset of UMLS provided the best representation, we also adopt this approach [7].

### 4.3 Graph-based Concept Weighting Models

**concepts-graph:** We apply the graph-based weighting method to concepts, as described in Section 3.1. We use the same parameter settings as terms-graph for $\phi$, iterations and context window. This model is denoted concepts-graph.

**concepts-graph-snomed:** Background information, derived from the SNOMED CT ontology, is injected into the concepts-graph weighting as described in Section 3.2, maintaining the same parameter settings. This model is denoted concepts-graph-snomed.

### 4.4 Evaluation Topics & Metrics

Evaluation was performed using the 34 topics from the TREC MedTrack'11 collection. Retrieval results were evaluated using Bpref and Precision @ 10 in accordance with the measures from TREC MedTrack'11. Bpref is regarded as the primary metric by MedTrack'11 and was used as the objective measure to tune the baselines terms-tfidf and concepts-tfidf.

### 4.5 Results

Retrieval results of the three baselines and the two graph-based concept methods are reported in Table 2.

| Run | Bpref | Prec@10 |
|---|---|---|
| terms-tfidf | 0.4722 | 0.4882 |
| concepts-tfidf | 0.4993 | 0.5176 |
| terms-graph | 0.4393 | 0.4882 |
| concepts-graph | 0.5050 (+15%) | 0.5441 (+11%) |
| concepts-graph-snomed | **0.5245** (+19%) | **0.5559** (+14%) |

**Table 2: Retrieval results on TREC MedTrack'11 using both term and concept representations, and after applying graph-based weighting and injection of domain knowledge. Percentage improvement shown over terms-graph.**

Comparing the term and concepts runs (terms-tfidf vs. concepts-tfidf), the concept based representation demonstrates improved performance. Comparing the effect of the graph-based weighting on terms (terms-tfidf and terms-graph) we actually observed degraded performance. However, when concepts are used to construct the graph (concepts-tfidf and concepts-graph), performance improved. The injection of domain knowledge using SNOMED CT (concepts-graph-snomed) provided additional improvements over concepts-graph in both bpref and precision. Analysis of results is presented in the next section.

Statistical significance using paired t-test was not found for any of the above results. The test collection contained only 34 query topics; van Rijsbergen comments that paired t-test may not reliably indicate statistical significance with small query sets [15]. Ideally, a larger query set or additional test collections would have been used; however, the medical domain does not currently have the diversity of evaluation resources available to other domains.

## 5. DISCUSSION

First, we consider the effect of using a bag-of-concepts rather than a bag-of-words representation — comparing the concepts-tfidf and terms-tfidf baselines. The use of a concept-based representation provides a 5% increase in bpref and 6% increase in P@10. This result is inline with previous concept-based approaches [7] and is encouraging for applying graph-based weighting to concept-based representations.

The effect of Blanco & Lioma's graph-based *term* weighting is now considered. When comparing the terms-tfidf and terms-graph baselines we observe that the use of graph weighting actually degraded retrieval performance by 6%. This result is contrary to the findings of Blanco & Lioma [3], who report improvements using the graph model on a number of test collections (over both tf-idf and BM25 baselines). Their corpora were newswire articles, web and blog crawls. The graph-based term weighting method may not be as suited to the peculiarities of medical IR; further analysis would be required to fully understand the reason for this.

In contrast to using terms, applying graph-based weighting to concepts *does* improve performance. Our concepts-graph model shows improvements over both the terms-tfidf and concepts-tfidf baselines, especially in precision, which exhibits an 11% improvement over the tuned terms-tfidf baseline and a 5% improvement over the tuned concept-tfidf baseline. Graph-based weighting is effective when using concepts, but not so when using terms. We hypothesise that this may be due to the fact that the concept representation encapsulates important medical n-grams as a single vertex in the graph (such as the Peptic Ulcer example from the concept graph of Figure 2). In contrast, the term-based graph does not encode these n-grams: instead, the two terms are split as separate vertices, both receiving a lower weight.

Overall, both the graph-based concept weighting methods (concepts-graph and concepts-graph-snomed), outperform the other three baselines in both bpref and precision @ 10. Although the small topic set makes statistical significance judgements difficult, we can provide some insights by considering how many queries were improved (and by how much) when using our concept graph method. Figure 5 show the change in bpref for each query using the concept-graph-snomed model when compare against the terms-graph baseline; topics ordered in decreasing change in bpref. The figure shows what

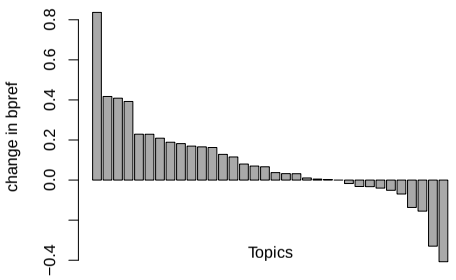



**Figure 4: Per-query change in bpref for concept-graph-snomed against terms-graph baseline.**

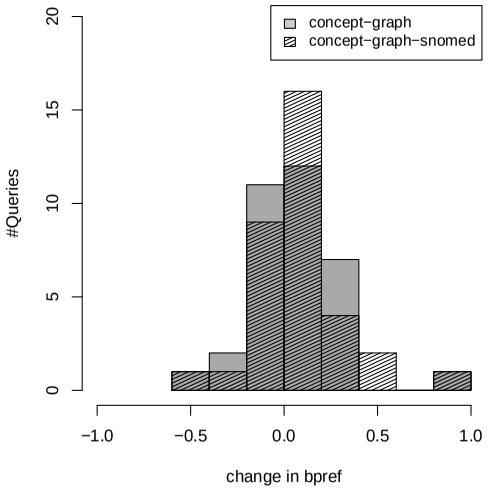



**Figure 5: Histogram showing #queries exhibiting change in bpref over term-graph for both concept graph models. Results show concepts-graph-snomed tends to make more small improvement to many queries — an indicator of increased robustness.**

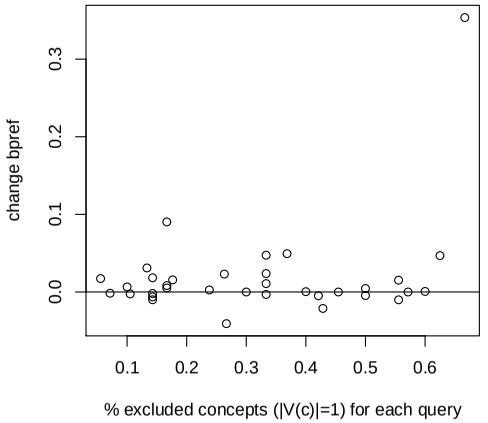



**Figure 6: The change in bpref when excluding query concepts with only one edge in the SNOMED CT graph. x-axis indicates the percentage of concepts for a given query where $|\mathcal{V}_s(c)| = 1$ (and are therefore excluded).**

a significance test should show — that change is seen in most queries, and change is for the better in most cases.

When comparing concept-graph-snomed to concept-graph, the injection of domain knowledge using SNOMED CT into the weighting provides an improvement in both bpref (4%) and precision (2%). Although the overall performance after injecting domain knowledge is not considerably higher, the injection method does provide some additional robustness across the query set. To illustrate this, Figure 5 shows the number of queries exhibiting change in bpref over the terms-graph baseline for both concept graph models. The histogram shows that concept-graph-snomed tends to make small variations (gains and losses) to a larger number of queries, whereas the concepts-graph has larger variations on a smaller number of queries. The former (small gains on many queries) indicates increased robustness and is more desirable for the model's general applicability. Both graph concept models do have the promising potential to benefit some queries substantially. Further study is need to enhance this aspect.

We now consider some interesting characteristics of the injection of domain knowledge. From Equation 7, the weighting of concept $c$ is dependent on the logarithm of the number of edges adjacent to $c$ in the SNOMED CT graph. Note, that when a concept has only one adjacent edge in the SNOMED CT graph, then the weight $w_b$ of query concept $c$ for document $d$ is zero ($\log|\mathcal{V}(c)| = \log 1 = 0$). In practice, this means that query concepts that contain only one edge in SNOMED CT are essentially ignored (their weight always being 0). Intuitively, this seems an undesirable characteristic that could lead to significant degradation in performance. To understand the extend of this characteristic and how it actually affects performance, we first consider how many queries contain concepts with only one edge in SNOMED CT (and therefore had scores of 0). The 34 test queries contained 448 concepts in total; of these a total of 127 (28%) had only one edge in the SNOMED CT graph, and were therefore ignored. Intuitively, ignoring so many concepts in the topic set would have a drastic effect on retrieval performance; however empirical results show the contrary. This is confirmed by Figure 6, which compares the change in bpref after applying the SNOMED CT weighting against the percentage of concepts within a given query where $|\mathcal{V}(c)| = 1$. Points on the far right of the x-axis indicate queries where many concepts were excluded from the weighting function. Note, that every query has at least one query concept excluded after applying SNOMED CT weighting. In addition, even when large portions of concepts are excluded from the query (far right of the x-axis) there are still positive changes in bpref. These queries contained a large number of concepts which were deemed as peripheral to the medical domain and, when excluded, aided performance.

Rather than completely exclude concepts with $|\mathcal{V}_s(c)| = 1$ we did perform experiments with alternative approaches that instead of excluding the concept, simply assigned a logarithmic scaled weight (e.g., $1 + log(|\mathcal{V}_s(c)|)$ or $log(1 + |\mathcal{V}_s(c)|)$). However, the best results in bpref were obtained when query concepts with only one adjacent edge in SNOMED CT were completely excluded. We conclude that a concept's lack of connectedness to other concepts in the domain ontology indicates they provide no additional information for the query and, in fact, may be misleading.

The exclusion of certain concepts based on the SNOMED CT connectedness is in effect a form of *query reduction*. Query reduction has been considered by researchers in information retrieval; finding an ideal subset of query terms can result in substantial performance gains [8, 2]. Kumaran

& Carvalho adopted a learning to rank approach that used statistical predictors (such as IDF, tf, Mutual Information and Query Clarity) to find an optimal query subset — they found an upper bound of 30% increase in performance, but their predictors only provided an 8% increase [8]. Bendersky & Croft [2] made use of corpus based statistics (such as IDF) and corpus independent indicators, such as Google n-grams, to identify and weight 'key concepts' within the query. They show improvements in retrieval, but found no robust feature across different test collections. We have shown that the use of a concept's connectedness in the SNOMED CT ontology provides an indicator of importance; in practice, providing a good feature for the implementation of an implicit query reduction method. Unlike previous approaches, our method used only one feature and avoided the use of heavy-weight machine learning to find an optimum feature combination; we also introduce no additional parameters. An interesting avenue of future work from this study is to consider query reduction specific to medical information retrieval, especially given the rich amount of domain knowledge available in resources such as SNOMED CT.

Finally, the findings of this study are applicable outside of the medical domain, specifically the injection of domain knowledge representing the importance of a term outside of the corpus being indexed. We currently use connectedness in SNOMED CT as the indicator of importance. Alternative weighting could be applied based on connectedness within any other resource represented as a graph, including domain specific resources, or general resources such as WordNet.

## 6. CONCLUSION

This paper presents a graph-based method to weight medical concepts found in documents for the purpose of medical IR. Graph-based representations are chosen over bag-of-words representations because they capture the relationships that exist between concepts, a feature important for capturing the innate dependencies in medical natural language. Additionally, concept-based representations are used to overcome vocabulary mismatch and to encapsulate important n-grams into a single concept.

We adapt previous graph-based term weighting method and apply them to concepts; a concept's weight is based on its PageRank score within the document. In addition, we present a novel method for the injection of domain knowledge regarding the concept's importance within the wider medical domain (not just the corpus itself). This method has an interesting characteristic of excluding a large number of query terms, resulting in a form of query reduction, and surprisingly leads to improvements in performance.

Evaluation was done on the TREC Medical Records track and a number of strong baselines were provided for comparison. Results showed that our graph-based concept weighting method outperforms each of the baselines.

The graph-based concept weighting method offers a framework for integrating formal background knowledge, often locked in medical domain ontologies, into data-driven approaches typical of information retrieval.

## 7. REFERENCES


[1] A. R. Aronson and F.-M. Lang. An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236, 2010.

[2] M. Bendersky and W. B. Croft. Discovering key concepts in verbose queries. In *Proceedings of the 31st annual International ACM SIGIR conference on research and development in information retrieval (SIGIR)*, pages 491–498, New York, NY, USA, 2008. ACM.

[3] R. Blanco and C. Lioma. Graph-based term weighting for information retrieval. *Information Retrieval*, 15(1):1–39, 2012.

[4] T. E. Doszkocs, J. Reggia, and X. Lin. Connectionist models and information retrieval. *Annual review of information science and technology*, 25:209–262, 1990.

[5] O. Egozi, S. Markovitch, and E. Gabrilovich. Concept-Based Information Retrieval using Explicit Semantic Analysis. *ACM Transactions on Information Systems*, 29(2):1–38, 2011.

[6] W. Hersh. *Information retrieval: a health and biomedical perspective*. Springer Verlag, New York, 3rd edition, 2009.

[7] B. Koopman, P. Bruza, L. Sitbon, and M. Lawley. Towards Semantic Search and Inference in Electronic Medical Records: an approach using Concept-based Information Retrieval. *Australasian Medical Journal: Special Issue on Artificial Intelligence in Health*, 5(9):482–488, 2012.

[8] G. Kumaran and V. R. Carvalho. Reducing long queries using query quality predictors. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 564–571, NY, USA, July 2009. ACM.

[9] Z. Liu and W. W. Chu. Knowledge-based query expansion to support scenario-specific retrieval of medical free text. *Information Retrieval*, 10(2):173–202, Jan. 2007.

[10] A. N. Nguyen, M. J. Lawley, D. P. Hansen, R. V. Bowman, B. E. Clarke, E. E. Duhig, and S. Colquist. Symbolic rule-based classification of lung cancer stages from free-text pathology reports. *Journal of the American Medical Informatics Association*, 17(4):440–445, 2010.

[11] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: bringing order to the web. *Technical Report, Stanford Digital Library Technologies*, 1999.

[12] C. Patel, J. Cimino, J. Dolby, A. Fokoue, A. Kalyanpur, A. Kershenbaum, L. Ma, E. Schonberg, and K. Srinivasclass. Matching patient records to clinical trials using ontologies. *The Semantic Web*, 4825:816–829, 2007.

[13] D. Ravindran and S. Gauch. Exploiting hierarchical relationships in conceptual search. In *Proceedings of the 13th annual international ACM CIKM conference on information and knowledge management*, pages 238–239. ACM, 2004.

[14] H. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3):187–222, July 1991.

[15] K. van Rijsbergen. *Information Retrieval*. Butterworth & Co, London, 2 edition, 1979.

[16] E. M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval*, pages 61–69, Dublin, Ireland, 1994. ACM.

[17] E. M. Voorhees and R. M. Tong. Overview of the TREC 2011 Medical Records Track. In *Proceedings of the Twentieth Text REtrieval Conference (TREC 2011)*, Gaithersburg, Maryland, USA, Nov. 2011.

[18] C. Zhai. Notes on the Lemur TFIDF model. Technical report, School of Computer Science, Carnegie Mellon University, 2001.

[19] W. Zhou, C. Yu, N. Smalheiser, V. Torvik, and J. Hong. Knowledge-intensive conceptual retrieval and passage extraction of biomedical literature. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval*, pages 655–662, New York, USA, 2007. ACM.

# A Study in Language Identification

Rachel Mary Milne
University of Otago
Dunedin, New Zealand
rmilne@cs.otago.ac.nz

Richard A. O'Keefe
University of Otago
Dunedin, New Zealand
ok@cs.otago.ac.nz

Andrew Trotman
University of Otago
Dunedin, New Zealand
andrew@cs.otago.ac.nz

## ABSTRACT

Language identification is automatically determining the language that a previously unseen document was written in. We compared several prior methods on samples from the Wikipedia and the EuroParl collections. Most of these methods work well. But we identify that these (and presumably other document) collections are heterogeneous in size, and short documents are systematically different from large ones. That techniques that work well on long documents are different from those that work well on short ones. We believe that improvement in algorithms will be seen if length is taken into account.

## Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Linguistic Processing; H.3.3 [**Information Search and Retrieval**]: clustering; I.2.7 [**Natural Language Processing**]: Language models

## General Terms

Experimentation

## Keywords

Language identification

## 1. INTRODUCTION

Almost anything you might want to do with a natural language document requires that you know what language it is in, even determining whether "1,234" is larger or smaller than 10. If you want to return documents relevant to a query, documents the user cannot read are not relevant.

It would seem that this problem could be solved at the source in many cases. Most text-holding elements in HTML, [11], for example, may have a "`lang`" attribute with an RFC 1766 [1] language code as value. XML [3] builds this into the XML framework as "`xml:lang`" [3, section 2.12]. Word processors such as Word and Symphony allow the author to set the language of any region of text.

This kind of annotation permits fine-grained classification. A document such as Olivier Lauffenburger's "Hittite Grammar" [7], for example, has raisins of Sumerian and Akkadian as well as plums of Hittite in a large custard of English.

However, such annotations are often omitted or misused. A document written in New Zealand English, for example, may be left with the word processor's default setting of American English (en_US), to the detriment of spelling and grammar checkers. Phillip Koehn [6] reported, for example, of the EuroParl corpus that "part of the 'English' part of the proceedings contain[ed] actually French texts [in May 1996]".

The language identification problem, then, is to automatically determine from the text itself what language it is written in. This problem has been addressed many times in the literature, but we uniquely identify that non-textural characteristics of a document can affect the accuracy of the algorithms. Specifically, we identify that the two corpora we use, Wikipedia and EuroParl, have multi-modal length distributions and this affects result quality. Short documents are obviously *harder* to classify just because they provide less evidence, but it turns out that they use language differently from long documents, so different algorithms are needed for each case.

It is obviously impossible to identify documents written in a language you have no knowledge of. It is also obviously difficult to discriminate between closely related languages given short documents: "The cat sat on the mat" is perfectly good American and perfectly good English. It is also clearly difficult to do fine-grained automatic classification, because of accidental similarities between languages ("come" is both Italian and English) and borrowings ("Matariki" is Māori, but is used in New Zealand English). What we can realistically hope for is automatic classification of whole documents into one of a small group of not too similar known languages.

## 2. RELATED WORK

Language identification is a well studied problem and space requirements prevent a thorough literature survey. However, several prior algorithms are discussed in this section.

Cavnar & Trenkle [4] developed an $n$-gram based text classifier, which they used to classify Usenet articles in English,



*ADCS '12,* December 05 - 06 2012, Dunedin, New Zealand
Copyright 2012 ACM 978-1-4503-1411-4/12/12...$15.00.

Portuguese, French, German, Italian, Spanish, Dutch and Polish. Here an $n$-gram is a contiguous subsequence of $n$ characters. For each language, they create a list of the common most $n$-grams in descending order of frequency, called an $n$-gram profile. These profiles are mixed in $n$-gram length with $n$ ranging from 1 to 5 (for example, the word "the" contributes "`t`", "`h`", "`e`", "`th`", and "`he`" as well as "`the`".)

Cavnar & Trenkle compare the $n$-gram profile of a new document with the $n$-gram profiles of the known languages, summing the absolute differences of the ranks ascribed to each $n$-gram. For example, if "the" is rank 1 in English and rank 9 in the unknown document, the absolute rank difference is 8. The language with the lowest sum is reported as the class of the new document. They reported an accuracy of 99.8% using this method.

Hayati [5] applied Cavnar & Trenkle's algorithm to a collection of Web documents in Danish, German, English, Spanish, Finnish, French, Italian, Dutch, Norwegian, Portuguese and Swedish. She found the accuracy to be lower than reported, at 86.8%. She suspected that their technique did not choose representative $n$-grams with sufficient power to distinguish between similar languages, so she used the Fisher discriminant function to choose $n$-grams and cosine similarity to compare document profiles to language profiles, and accuracy improved to 93.9%.

Langdetect [12] is an open-source Java library for language identification. It uses a naïve Bayes algorithm with character $n$-grams. McCandless [9] compared three classifiers, of which langdetect was the best. He reported an accuracy of 99.2% on a collection with 17 languages, ranging from 97.2% for Danish to 100.0% for Greek. Langdetect comes with built-on profiles, so we did not train it in our experiments. However we used our own markup removal algorithm, as it does not do this itself, Langdetect reports several possibilities; we only used the language with the highest score.

Mayer [8] looked at tweets and e-Bay messages, which are very short. He used the first two and last two words in each document and looked them up in dictionaries for each language.

It is Mayer's work that first alerted us to the possibility that different algorithms might perform better on different length documents. To this end we analysed the length of documents in the collections we used and identify that they are mixed-modal. We consequently tested several algorithms on these different lengths and show here that, indeed, different algorithms are suited to different lengths of document.

# 3. LANGUAGES USED

This study primarily worked with four languages: German (de), English (en), Spanish (es), and French (fr). They were chosen because one author was able to read all these languages and consequently articulate why the classifiers were failing (recall that Koehn reported misclassified documents in EuroParl). In section 6 Dutch and Italian were further examined.

<sing languages that could be written in the ISO Latin 1 character set simplified the coding but also makes the identification problem harder. For this set of languages, conversion to lower case is language-independent. Equally, separating Chinese from Russian from English can be done by mechanically examining the codepage used in the majority of the document.

# 4. DOCUMENT COLLECTIONS USED

We worked with two collections: the Wikipedia [13] and the EuroParl collection [6].

September 2012 Wikipedia dumps were obtained for German, Dutch, English, Spanish, Italian, and French. They were decompressed, parsed as XML, the <text> elements were extracted, Wikimedia markup was removed, and they were tokenised. Stemming and stopping were not performed because they cannot be performed until the language is known. Wikimedia markup removal was done with *ad-hoc* code. Words were converted to lower case. The results of section 6 were obtained using one author's Wikimedia stripper, and randomly selected training sets and test sets of 100,000 documents each for each language, only documents with more than 10 words being selected. The results of section 8 were obtained using another author's Wikimedia stripper, and a training set of 1,000 documents in each language. Although we used two different strippers we do not believe that this substantially affects the results. Equally, we do not believe that the different numbers of test documents will substantially affect results either.

The mid-2012 edition of the EuroParl collection was obtained, decompressed, and the XML markup removed. Stemming and stopping were not done (recall that they are language dependent). Words were converted to lower case. Test sets of 1,000 documents were used. In section 6 only documents with more than 10 words were selected.

Document lengths were also checked in the Wall Street Journal and INEX IEEE article corpora.

# 5. NONUNIFORMITY OF COLLECTIONS

Our preliminary investigatory experiments suggested that each technique performed worse than expected. Closer inspection showed that most of the difficulty was with very short documents.

Figure 1 shows a histogram of (log Wikipedia article length in words + 1) for each of German, English, Spanish, and French. The overlaid curves show normal distributions fitted to the data using the `normalmixEM` function from the "mixtools" package [2] in R [10]. This simple mixture distribution appears from visual inspection to fit well. The fact that the document length *distribution* can be modelled as a mixture of two simpler distributions strongly suggests that the *collection* is actually a mixture of two collections with different properties.

If one of these distributions were small relative to the other the it might be effective to simply ignore the bi-modality of the collection. Table 1 shows the proportion of documents that fell into the "small" (about 2 words) group and proportion that fell into the "large" group (averaging about 160 words) for each of the four languages. Neither group is negligible.

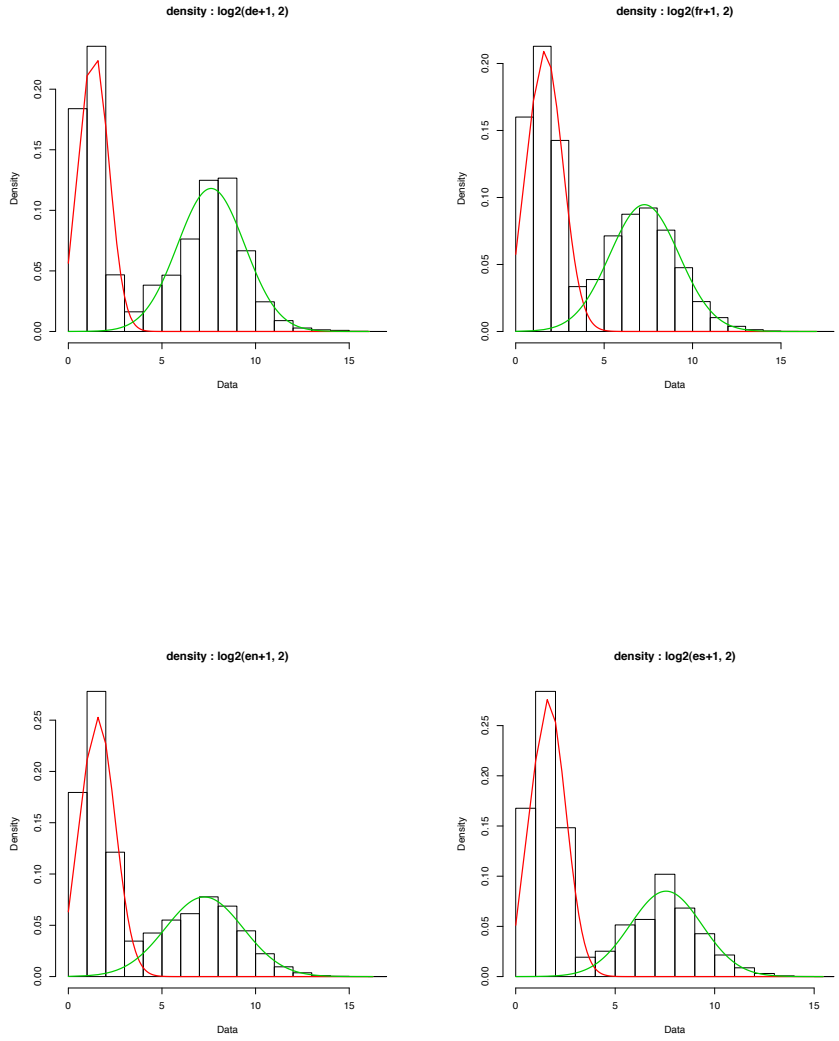



**Figure 1: Length distribution of Wikipedia articles showing a bi-modal distribution in all four languages examined**

Figure 2 shows the equivalent histogram (log document length in words + 1 against proportion of documents) for the same four languages, in the EuroParl collection. Again the overlaid curves show normal distributions fitted using `normalmixEM`. This time, a mixture of four normals fitted best (by visual inspection).

Table 2 shows the proportion of EuroParl documents that fell into the "small" (about 10 words), "medium" (about 170 words), "large" (about 8,000 words), and "huge" (about 80,000 words) groups for each of the four languages.

We found other collections to be mixed as well. For example, the INEX IEEE collection appears to be a mixture of three groups (about 800, about 1900, and about 5000 words). In further experiments we will examine further collections to determine whether this is a pattern we can expect of whether it is characteristic of just these collections.

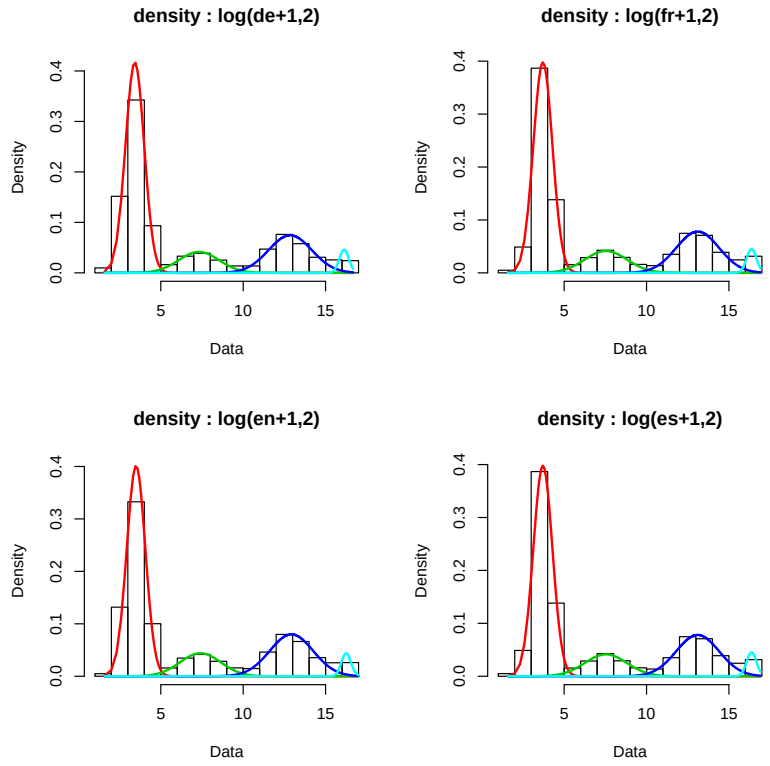



**Figure 2: Length distribution of EuroParl documents showing a quad-modal distribution in all four languages examined**

| Language | Small | Large |
|---|---|---|
| de | 47% | 53% |
| en | 59% | 41% |
| fr | 53% | 47% |
| es | 61% | 39% |

**Table 1: Proportion of Wikipedia documents of each modality for each of the four languages**

| Language | Small | Medium | Large | Huge |
|---|---|---|---|---|
| de | 60% | 13% | 24% | 3% |
| en | 57% | 14% | 26% | 3% |
| fr | 58% | 13% | 26% | 3% |
| es | 58% | 14% | 24% | 3% |

**Table 2: Proportion of EuroParl documents of each modality for each of the four languages**

We observe that short documents tend to use language differently from long ones and very short documents more so. In the EuroParl collection we observe headlines or stylised "see XXX committee minutes". Similarly within the Wikipedia, redirect articles are common. Table 3 shows that *as a whole*, short Wall Street Journal articles do not look like longer ones. When judging the effectiveness of language classifiers, we believe it is important to judge the size classes separately. A language identification may do well on one size class and badly on another.

| Top 20 words | | | Top 20 trigrams | | |
|---|---|---|---|---|---|
| Small | Medium | Large | Small | Medium | Large |
| of | the | the | of | the | the |
| a | of | of | a | ion | ing |
| the | a | to | the | of | of |
| from | to | a | ent | ing | to |
| in | and | in | ion | a | ion |
| and | in | and | ing | and | and |
| to | said | s | and | to | a |
| was | s | that | ear | ent | in |
| said | million | for | ice | ill | ent |
| this | for | is | rom | in | tio |
| president | it | on | in | com | for |
| year | its | it | fro | tio | s |
| named | that | as | res | for | tha |
| director | from | at | to | lli | ter |
| earlier | company | with | was | res | hat |
| s | is | by | ect | aid | ate |
| vice | will | said | ill | ter | ati |
| for | by | mr | inc | sai | ill |
| million | on | he | ati | lio | ers |
| board | year | from | aid | ati | com |

**Table 3: Wall Street Journal; words and trigrams; small = 1 to 30 words, medium = 31 to 300 words; large = over 300 words.**

| | de | nl | en | es | it | fr |
|---|---|---|---|---|---|---|
| de | 99.4% | 0.1% | 0.3% | 0.0% | 0.1% | 0.1% |
| nl | 0.4% | 99.3% | 0.2% | 0.0% | 0.1% | 0.0% |
| en | 0.7% | 0.5% | 98.4% | 0.1% | 0.3% | 0.0% |
| es | 0.3% | 1.5% | 0.5% | 97.4% | 0.3% | 0.1% |
| it | 1.3% | 0.1% | 9.4% | 1.5% | 87.6% | 0.1% |
| fr | 0.7% | 0.5% | 0.5% | 0.3% | 0.4% | 97.5% |

**Table 4: Wikipedia confusion matrix, $k = 20$. Row = true language, column = assigned language.**

| | de | nl | en | es | it | fr |
|---|---|---|---|---|---|---|
| de | 99.5% | 0.1% | 0.3% | 0.1% | 0.0% | 0.0% |
| nl | 0.1% | 99.3% | 0.3% | 0.2% | 0.0% | 0.0% |
| en | 0.3% | 0.1% | 99.3% | 0.1% | 0.1% | 0.1% |
| es | 0.1% | 0.1% | 0.6% | 99.1% | 0.0% | 0.1% |
| it | 0.1% | 0.1% | 1.3% | 8.0% | 90.4% | 0.1% |
| fr | 0.2% | 0.1% | 1.0% | 0.3% | 0.1% | 98.3% |

**Table 5: Wikipedia confusion matrix, $k = 1000$. Row = true language, column = assigned language.**

# 6. BASELINE

It is reasonable to conclude from the document length analysis and literature survey that collections may not mixed, and that if it were possible to identify which component a document belongs to then it might be possible to increase the performance of language identification. This, in turn, might (although we leave it for further work to demonstrate this) increase the precision of a search engine.

In order to get a clear view of the merits of each method when applied to long documents (Documents shorter than 11 words were excluded), we implemented a simple term-based baseline.

A language profile was built for each language by taking the top $k$ most frequent word from the training set for that language. For each document in the test set, the unique words were identified and compared to these language profiles. This comparison was a straight set intersection. The language of the document was chosen as the language of the largest set. That is, if the document contained 23 of the top 50 words in English but only 12 of the top 50 words in German then the document was identified as bring written in English.

This approach resembles that of Cavnar & Trenkle, but pays no attention to relative ranks. In later sections we refer to this method (with $k = 1000$) as the "top 1000 words" method.

Table 4 presents the confusion matrix for the Wikipedia test set, trained on the Wikipedia training set when $k = 20$. Cells were rounded to the nearest 0.1%. It is reasonable to expect more evidence to produce better results and so the experiment was re-performed with $k = 1000$ (see Table 5). Percentages are presented (rather than absolute counts) in these tables so that they can be more easily compared.

It is astonishing that such a crude technique performs so well (typically over 95% accuracy). Further analysis is required to explain why, however we believe that it is due to a combination of Zipf's law and Heap's law — that is, the most frequent words will be frequent and we're not really expecting to see many new ones in a new document

We make further observations from these results: if Italian had not been included, the results would have been better; Italian is often mistaken for Spanish; and everything is mistaken for English.

These observations highlight two difficulties in language identification of European languages. First, Italian genuinely has a number of high frequency words which match English and Spanish words, so an approach such as Cavnar & Trenkle's absolute rank difference or Kulback-Liebler divergence should improve results. Second, it is difficult to accurately remove all markup and any remaining Wikimedia markup will be identified as "English", as markup is frequent this adds to the confusion.

An error analysis on by document size suggests that most errors in classifying Italian occurred in documents of between 121 and 175 words in length where about 40% were wrongly classified.

To eliminate any error introduced as a consequence of markup a further experiment was conducted using the EuroParl collection. In this experiment the two collections were tested against themselves and each other. Rather than presenting more confusion matrices, Table 6 shows the accuracy (percent of correctly classified documents) we observed when $k = 20$. The last line is the main diagonal of Table 4 represented for clarity.

This table shows that the classifiers developed for each collection work well on the other. Indeed, the column for Italian suggests that the baseline method effectively manages markup contamination in the training documents and the test documents, but suffers when both are contaminated the same way.

Increasing $k$ does not always improve (sometimes worsening) accuracy as table 7 for $k = 200$ shows. As before, contamination of the Wikipedia data resulting from inferior Wikimedia markup removal is a problem.

The purpose of this experiment was to establish a *lower bound* that better methods should beat. What we found is that data cleansing (removal of markup from XML, removal of menus from Web pages, and so on) can substantially affect the performance of the language detection algorithm. This happens because the markup tends to be in one language, but becomes frequent in all languages making all languages look (to the classifier) more similar than they really are.

# 7. TRIGRAMS

We conducted two experiment with tri-grams. In the first we examined which tri-grams were frequent, and in the second we measured the performance of the approach.

The algorithm we used, "Padded trigrams", is a reduced version of the Cavnar & Trenkle algorithm using only tri-grams (recall that they used 1-5 grams). Words that were shorter

| Training | Testing | de | nl | en | es | it | fr |
|---|---|---|---|---|---|---|---|
| EuroParl | EuroParl | 99.4% | 99.5% | 99.5% | 98.3% | 98.0% | 96.7% |
| EuroParl | Wikipedia | 99.5% | 99.3% | 97.9% | 97.8% | 94.1% | 97.4% |
| Wikipedia | EuroParl | 99.5% | 99.5% | 99.5% | 98.4% | 96.4% | 96.6% |
| Wikipedia | Wikipedia | 99.4% | 99.3% | 98.4% | 97.4% | 86.7% | 97.5% |

**Table 6: Accuracy for each language, varying training and test collections, $k$=20**

| Trained | Tested | de | nl | en | es | it | fr |
|---|---|---|---|---|---|---|---|
| EuroParl | EuroParl | 99.5% | 99.6% | 99.8% | 99.8% | 100.0% | 99.4% |
| EuroParl | Wikipedia | 99.4% | 99.3% | 98.4% | 97.5% | 88.6% | 97.6% |
| Wikipedia | EuroParl | 99.6% | 99.1% | 96.0% | 98.5% | 99.5% | 97.4% |
| Wikipedia | Wikipedia | 99.5% | 99.4% | 98.6% | 99.2% | 90.1% | 98.1% |

**Table 7: Accuracy for each language, varying training and test collections, $k$=200**

than three grams were padded with space (on the right). For example, "a bag of meal" contributes "a␣␣", "bag", "of␣", "mea", and "eal".

Table 8 shows the top ten trigrams for the Wikipedia collection in four languages. Articles and conjunction are high in all four lists. The trigrams "ing", "ion", "tio", and "ent" are characteristic of nominalisations in English; there are hints of the same pattern in the other languages. This suggests that the Wikipedia may not be typical of language use. Also seen in this list are some highly frequent words (for example the articles, "the" and "a" in English).

Table 9 shows the top ten trigrams for the EuroParl collection. While the trigrams are different, a similar pattern is seen.

The presence of whole words in the top tri-gram lists may be providing additional reason for the success of the baseline approach in the previous section (and *vice versa*). Highly frequent words (at least in English) appear to be short and the short words appear to be frequent tri-grams.

To examine the performance of this algorithms, the Wikipedia collection was split into three groups: 1000 randomly chosen short documents (10 words or fewer) for testing; 1000 randomly chosen long documents (more than 10 words) for testing; and the remainder used for training. All of the EuroParl articles were used for testing.

As this algorithm can result in ties, a document is reported as "Tie" if two or more languages tied for best score.

Table 10 shows the confusion matrix for the trigram method applied to long Wikipedia documents. As before, rows show the language ascribed to a document in its collection and columns show the language it was classified as. Cells are percentages. A large percentage on the main diagonal is good and non-zero results off the diagonal show mistakes being made. The results for long Wikipedia documents are generally good. Table 11 shows what happens with short Wikipedia documents. The results are generally not good. Table 12 shows the confusion matrix for long EuroParl documents. The results are good. Table 13 shows what happens with short EuroParl documents. The results are also not good, but they are better than the results for short

| en | de | es | fr |
|---|---|---|---|
| 2.36% the | 1.31% sch | 2.31% de | 1.75% de |
| 1.00% and | 1.21% der | 1.09% la | 0.96% ent |
| 1.00% of | 1.05% ein | 0.93% en | 0.88% la |
| 0.85% ing | 0.98% ich | 0.86% el | 0.84% ion |
| 0.78% ion | 0.88% che | 0.84% ent | 0.68% le |
| 0.73% in | 0.80% die | 0.72% y | 0.67% que |
| 0.63% to | 0.76% und | 0.64% con | 0.65% les |
| 0.62% a | 0.61% den | 0.63% nte | 0.64% et |
| 0.58% tio | 0.59% ter | 0.61% que | 0.61% tio |
| 0.56% ent | 0.58% ung | 0.60% ado | 0.59% à |

**Table 8: Top 10 trigrams by language: Wikipedia**

| en | de | es | fr |
|---|---|---|---|
| 3.04% the | 1.56% ich | 2.03% de | 1.59% de |
| 1.32% ion | 1.44% die | 1.38% la | 1.58% ent |
| 1.17% of | 1.42% der | 1.22% que | 1.40% ion |
| 1.08% to | 1.20% sch | 1.17% ent | 1.08% que |
| 1.03% and | 1.12% ein | 1.02% ión | 1.04% la |
| 0.93% ent | 1.03% ung | 1.02% est | 0.96% tio |
| 0.89% tio | 0.99% che | 0.91% nte | 0.87% ons |
| 0.89% ing | 0.92% den | 0.88% en | 0.87% men |
| 0.77% in | 0.83% und | 0.85% con | 0.86% les |
| 0.66% hat | 0.83% cht | 0.79% el | 0.76% l |

**Table 9: Top 10 trigrams by language: EuroParl**

Wikipedia documents; short EuroParl documents tend to be longer than short Wikipedia ones.

# 8. EXPERIMENT

So far we have show that the document collections we are using are mixtures and the the performance of the baseline algorithms on short documents is substantially worse than on longer documents. In this section we show that this disparity of performance is not a characteristic of our baselines by comparing approaches of others on different length documents.

Four methods were evaluated: Cavnar & Trenkle's method, our padded trigrams method, langdetect, and the top 1000 words method. To save space, Tables 14–17 show accuracies, not entire confusion matrices. These tables show that on the Wikipedia collection the classifiers are effective on long

| | en | de | es | fr | Tie |
|---|---|---|---|---|---|
| en | 99.2% | 0.1% | 0.2% | 0.5% | 0.0% |
| de | 0.7% | 99.2% | 0.1% | 0.0% | 0.0% |
| es | 0.5% | 0.2% | 99.3% | 0.0% | 0.0% |
| fr | 1.3% | 0.0% | 1.3% | 97.4% | 0.0% |

**Table 10: Confusion matrix: Wikipedia long documents**

| | en | de | es | fr | Tie |
|---|---|---|---|---|---|
| en | 48.0% | 14.8% | 17.2% | 13.8% | 6.2% |
| de | 14.5% | 55.0% | 13.3% | 11.0% | 6.2% |
| es | 12.4% | 9.0% | 57.3% | 15.9% | 5.4% |
| fr | 13.1% | 9.5% | 17.0% | 56.0% | 4.4% |

**Table 11: Confusion matrix: Wikipedia short documents**

| | en | de | es | fr | Tie |
|---|---|---|---|---|---|
| en | 99.4% | 0.3% | 0.0% | 0.2% | 0.0% |
| de | 0.3% | 99.5% | 0.1% | 0.1% | 0.0% |
| es | 0.1% | 0.0% | 99.8% | 0.0% | 0.0% |
| fr | 0.2% | 0.1% | 0.0% | 99.7% | 0.0% |

**Table 12: Confusion matrix: EuroParl long documents**

| | en | de | es | fr | Tie |
|---|---|---|---|---|---|
| en | 74.9% | 9.9% | 1.0% | 14.2% | 0.0% |
| de | 0.6% | 99.0% | 0.2% | 0.3% | 0.0% |
| es | 0.2% | 4.5% | 94.9% | 0.5% | 0.0% |
| fr | 2.9% | 5.3% | 1.5% | 90.3% | 0.0% |

**Table 13: Confusion matrix: EuroParl short documents**

| Method | de | en | es | fr |
|---|---|---|---|---|
| Padded trigrams | 99.2% | 99.2% | 99.3% | 97.4% |
| Cavnar & Trenkle | 99.6% | 98.6% | 96.4% | 97.3% |
| langdetect | 99.5% | 97.1% | 96.2% | 97.4% |
| Top 1000 Words | 99.3% | 98.9% | 98.8% | 98.4% |

**Table 14: Accuracy of the four methods on Wikipedia long documents**

| Method | de | en | es | fr |
|---|---|---|---|---|
| Padded trigrams | 48.0% | 55.0% | 57.3% | 56.0% |
| Cavnar & Trenkle | 59.4% | 56.4% | 66.3% | 61.1% |
| langdetect | 67.0% | 54.7% | 69.8% | 67.0% |
| Top 1000 Words | 32.3% | 25.8% | 35.8% | 39.6% |

**Table 15: Accuracy of the four methods on Wikipedia short documents**

| Method | de | en | es | fr |
|---|---|---|---|---|
| Padded trigrams | 98.4% | 99.5% | 99.8% | 99.6% |
| Cavnar & Trenkle | 99.9% | 99.3% | 99.8% | 99.2% |
| langdetect | 99.4% | 99.6% | 99.8% | 99.6% |
| Top 1000 Words | 99.3% | 100.0% | 99.4% | 99.5% |

**Table 16: Accuracy of the four methods on EuroParl long documents**

| Method | de | en | es | fr |
|---|---|---|---|---|
| Padded trigrams | 74.9% | 99.0% | 94.9% | 90.3% |
| Cavnar & Trenkle | 93.0% | 88.1% | 95.2% | 89.4% |
| langdetect | 93.5% | 96.2% | 95.5% | 93.5% |
| Top 1000 Words | 91.6% | 94.6% | 90.6% | 91.9% |

**Table 17: Accuracy of the four methods on EuroParl short documents**

| Method | Len | de | en | es | fr |
|---|---|---|---|---|---|
| Cavnar & Trenkle | >10 | 98.4% | 99.7% | 96.3% | 96.2% |
| Padded trigrams | >10 | 99.3% | 99.3% | 97.1% | 97.8% |
| Cavnar & Trenkle | $\leq 10$ | 56.2% | 58.8% | 62.9% | 43.2% |
| Padded trigrams | $\leq 10$ | 56.8% | 40.3% | 54.2% | 52.3% |

**Table 18: Crossover accuracy, trained on EuroParl, tested on Wikipedia**

documents but not on short documents. A similar pattern is seen on EuroParl collection, however the accuracy on short documents on that collection degrades more gracefully.

To determine whether this is a collection-specific characteristic or not we performed one further experiment. In that experiment the classifiers were trained on one collection and tested on the other. Tables 18 and Table 19 show how well the algorithms performed when used in this way. It appears as though identifying the language of short Wikipedia documents is substantially harder than doing so for EuroParl documents.

# 9. CONCLUSIONS

In this work we examined two document collections and identified that documents do not follow a normal distribution in size, but are instead multi-modal in size. In the Wikipedia we identified two components (short and long) and observed that short documents were often redirect pages. The EuroParl corpus contained 4 components we called small, medium, large, and huge. In this collection short pages were similarly redirect pages; "see the minutes of" pages.

When we tested a simple baseline language identification technique we observed unexpectedly high performance. We also observed that performance was inhered by markup whose language model pollutes the models of each of the languages we were testing for.

Our experiments show that short documents may need different techniques from long ones. If such documents are redirect (or equivalent) pages then this suggests that an effective way of classifying short documents might be through the classification of the pages they point to. We leave for

| Method | Len | de | en | es | fr |
|---|---|---|---|---|---|
| Cavnar & Trenkle | >10 | 99.6% | 99.4% | 99.8% | 99.3% |
| Padded trigrams | >10 | 99.4% | 07.0% | 99.8% | 99.7% |
| Cavnar & Trenkle | $\leq 10$ | 93.1% | 94.2% | 94.5% | 91.0% |
| Padded trigrams | $\leq 10$ | 97.9% | 57.5% | 94.4% | 91.9% |

**Table 19: Crossover accuracy, trained on Wikipedia, tested on EuroParl**

further work the exploration of such an approach.

As we expected, some languages are more similar than others. The classifiers we used considered Italian and Spanish to be similar - indeed English variants may be extremely hard to distinguish as some documents could be correct both syntactically and semantically in more than one variant. We leave for further work the construction of a taxonomy of languages that might be used by a classifier to improve performance.

Finally we observe that word-based and character $n$-gram based classifiers respond to different aspects of a language — we leave for further work methods to combine them.

# 10. REFERENCES


[1] H. Alvestrand. Tags for the Identification of Languages. RFC 1766 (Proposed Standard), Mar. 1995. Obsoleted by RFCs 3066, 3282.

[2] T. Benaglia, D. Chauveau, D. R. Hunter, and D. Young. mixtools: An R package for analyzing finite mixture models. *Journal of Statistical Software*, 32(6):1–29, 2009.

[3] T. Bray, J. Paoli, and C. M. Sperberge-McQueen. Extensible markup language (xml) 1.0. REC-xml-19980210 (W3C recommendation), February 1998.

[4] W. B. Cavnar and J. M. Trenkle. $n$-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1994.

[5] K. Hayati. Language identification on the world-wide web. Master's thesis, Computer Science, University of California, Santa Cruz, 2004.

[6] P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT Summit 2005*, 2005.

[7] O. Lauffenburger. Hittite grammar, 2006. [online, last visited on October 2012].

[8] U. Mayer. Bootstrapped language identification for multi-site internet domains. In *Proceedings of KDD'12*, August 2012.

[9] M. McCandless. Accuracy and performance of google's compact language detector, October 2011. [blog entry; last visited in October 2012].

[10] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.

[11] D. Raggett, A. Le Hors, and I. Jacobs. Html 4.01 specification. REC-html40 (W3C recommendation), December 1999.

[12] N. Shuyo. Language detection library for java, 2010.

[13] Wikipedia. Wikipedia:database download, 2012.

# Efficient Indexing Algorithms for Approximate Pattern Matching in Text

Matthias Petri
School of of CS&IT
RMIT University and NICTA VRL
Melbourne, Victoria, 3000
matthias.petri@rmit.edu.au

J. Shane Culpepper
School of of CS&IT
RMIT University and NICTA VRL
Melbourne, Victoria, 3000
shane.culpepper@rmit.edu.au

## ABSTRACT

Approximate pattern matching is an important computational problem with a wide variety of applications in Information Retrieval. Efficient solutions to approximate pattern matching can be applied to natural language keyword queries with spelling mistakes, OCR scanned text incorporated into indexes, language model ranking algorithms based on term proximity, or DNA databases containing sequencing errors. In this paper, we present a novel approach to constructing text indexes capable of efficiently supporting approximate search queries. Our approach relies on a new variant of the Context Bound Burrows-Wheeler Transform ($k$-BWT), referred to as the Variable Depth Burrows-Wheeler Transform ($v$-BWT). First, we describe our new algorithm, and show that it is reversible. Next, we show how to use the transform to support efficient text indexing and approximate pattern matching. Lastly, we empirically evaluate the use of the $v$-BWT for DNA and English text collections, and show a significant improvement in approximate search efficiency over more traditional $q$-gram based approximate pattern matching algorithms.

## Keywords

Burrows-Wheeler Transform, Approximate Pattern Matching

## 1. INTRODUCTION

Approximate pattern matching is a classic problem in computer science with a wide variety of applications [10, 13]. For example, the role of approximate pattern matching in biological applications has been well documented [8]. Efficient solutions to approximate pattern matching can also be applied in a variety of Information Retrieval applications. Examples where approximate pattern matching can be applied in the IR domain include natural language keyword queries with spelling mistakes [10], OCR scanned text incorporated into indexes [10], language model ranking algorithms based on term proximity [12], or DNA databases containing sequencing errors [11].


*ADCS'12,* December 5–6, 2012, Otago, Dunedin, New Zealand.


The approximate pattern matching problem can be defined as follows: LOCATE, COUNT, or EXTRACT all occurrences of pattern $P$ of length $m$ in a text $T$ of size $n$ with at most $k$ errors. If $n$ is not large and only a few search queries will be performed, *on-line* algorithms such as `agrep` [26] can perform these operations in time proportional to the length of the text. However, on-line solutions are typically not sufficient for massive document collections, or situations which require a large number of queries to be run on the same collection. In these scenarios, building an index capable of supporting approximate matching queries is desirable. In this paper, we focus on a new approach to indexing and searching text collections allowing errors.

One viable approach to indexing text collections allowing errors is to use a modified self-index to support approximate text matching [23]. Most research in this domain has focused on providing worst case performance guarantees using a suffix array to perform fast substring matches [3]. In contrast, inverted indexes using $q$-grams can also be used, and generally perform well in practice despite providing no worst case performance guarantees. A $q$-gram index is simply an inverted index storing positions of all distinct substrings of length $q$ in $T$. The $q$-gram index is used as a filtering tool to generate potential positions in $T$ matching $P$. These positions must then be verified in the text using a variety of different *edit distance* based algorithms [17].

A weakness of traditional $q$-gram indexes is the use of fixed text segments of length $q$. If $q$ is small, the inverted files can be very long for common $q$-grams, degrading performance for many queries. However, if $q$ is large, then the size of the index grows at an unacceptable rate. Recently, Navarro and Salmela [18] show that using variable length $q$-grams can help find the best tradeoff between the length of the postings lists, and the total number of "grams" that must be indexed. Unfortunately, the approach to finding the substrings to be indexed still requires the creation of a suffix tree, which can dominate the construction time of the index.

### 1.1 Our Contribution

We present a new variant of the BWT, called the variable depth Burrows-Wheeler transform ($v$-BWT). We prove that the transform is always reversible, and show how it can be used to create a variable length $q$-gram partitioning for variable length $q$-gram indexes without requiring a complete suffix array. We describe how to use the $v$-BWT to construct a self-index, precluding the need to explicitly represent the $v$-grams using postings lists. We empirically evaluate the usefulness of our transform by comparing the number of verifications required when performing approximate matching on both DNA and English text. Finally, we describe future work where we intend to apply our new approach to common IR problems.

## 2. BACKGROUND AND RELATED WORK

We define a text $T[0..n-1]$ of $n$ symbols and a pattern $P[0..m-1]$ of $m$ symbols over an alphabet $\Sigma$ of size $\sigma$. We denote the symbol \$ to be lexicographically smaller than all symbols in $\Sigma$. Without loss of generality, we require \$ to be the last symbol in $T$. We refer to the BWT over $T$ as $T^{\text{BWT}}$, the $k$-BWT transformed text as $T^{k\text{-BWT}}$ and the variable depth transformed text as $T^{v\text{-BWT}}$. We define $\mathcal{M}$ to be a matrix containing all lexicographically sorted cyclic rotations of $T$. We similarly define $\mathcal{M}_k$ and $\mathcal{M}_v$ to be the equivalent matrix for the $k$-BWT and $v$-BWT.

### 2.1 Text Transformations

The BWT was originally proposed by Burrows and Wheeler [2] as the first step in a transform based compression system. The transform is used to permute $T$ so symbols with similar context are grouped together. Conceptually, the BWT creates a matrix $\mathcal{M}$ consisting of all rotations of $T$. The rows in the matrix are then sorted based on the lexicographical ordering. $T^{\text{BWT}}$ refers to the last column of $\mathcal{M}$. $T^{\text{BWT}}$ is usually more compressible than $T$. The BWT is reversible in $\mathcal{O}(n)$ time without the need of any additional information using the following steps: (1) Recover the first column $F$ of $\mathcal{M}$ by sorting $T^{\text{BWT}}$ in lexicographical order. (2) Compute the mapping between the first $F$ and last column $L$ so $LF(i) = j$ if $F[j] = L[i]$. (3) Using the $LF()$ mapping, we can recover $T$ from $T^{\text{BWT}}$ in reverse order as $T[i] = T^{\text{BWT}}[LF(j)]$.

One of the main problems of constructing the BWT is that individual row comparisons in $\mathcal{M}$, or the equivalent suffix sorting comparisons in a suffix array construction algorithm can be computationally expensive. Two suffixes are compared by iterating over $T$ starting from each respective position. In the worst case, each suffix comparison can take $\mathcal{O}(n)$ time. To alleviate this problem, Schindler [24] and Yokoo [27] independently proposed a bounded version of the BWT, the $k$-BWT. The $k$-BWT compares each row/suffix up to a depth of $k$ symbols while stable sorting the equal rows based on initial text positions. This guarantees that each suffix comparison can be done in $\mathcal{O}(k)$ time. However, this implies that two suffixes are considered equal if they share the same $k$-prefix in $\mathcal{M}$. All rows in $\mathcal{M}$ sharing the same $k$-prefix are grouped together in a context group. Within a context group, rows are sorted based on the corresponding position in $T$. This implies that the suffix array positions in each context group are in monotonically increasing order.

To recover $T$ from $T^{k\text{-BWT}}$, the boundaries of the context groups in BWT are required as LF() only returns the correct result if the BWT is fully sorted lexicographically. Definition 1 defines a bitvector $D_k$ describing the context group boundaries:

DEFINITION 1. *For any $0 \leq k < n$, let $D_k[0, n-1]$ be a bitvector, such that $D_k[0] = 1$ and, for $1 \leq i < n$,*

$$D_k[i] = \begin{cases} 0 & \text{if } \mathcal{M}_k[i][0,k-1] = \mathcal{M}_k[i-1][0,k-1] \\ 1 & \text{if } \mathcal{M}_k[i][0,k-1] \neq \mathcal{M}_k[i-1][0,k-1] \end{cases}$$

LF() is still guaranteed to jump to the correct context group in $\mathcal{M}$ corresponding to the previous symbol in $T$ as the individual $k$-groups are still sorted lexicographically [21]. Recall that with a context group, the rows are sorted based on the initial position in $T$. As $T$ is recovered in reverse sequential order, during the recovery of $T$, each $k$-group is processed in reverse sequential order. Fortunately, the context group boundaries ($D_k$) can be recovered from $T^{k\text{-BWT}}$ in $\mathcal{O}(n)$ time [20]. To recover $T$ from $T^{k\text{-BWT}}$, the $k$-group boundaries are recovered first. The $LF()$ mapping is then used to jump between context groups, while using $D_k$ to process each individual context group in reverse sequential order. Interestingly, although the additional cost of recovering the context group is asymptotically worse than in the full BWT, $T$ can be recovered faster from $T^{k\text{-BWT}}$ than from $T^{\text{BWT}}$ due to the sequential access of each context group. The sequential access results in a significant cache effect not present in the random jumps induced by the BWT [4].

### 2.2 Self-Indexing

The BWT has been used in many compression systems to increase the compressibility of the text. Additionally, the BWT is the core of many compressed indexing schemes as there exists a duality between the BWT and the suffix array: $SA[i] = T^{bwt}[i]-1$. The duality between $T^{\text{BWT}}$ and the suffix array over $T$ allows searching in $T$ using only compressed representation of $T^{\text{BWT}}$ [5]. This type of text index is usually referred to as a self-index as $T$ is not required to perform search. Self-indexes support the following operations efficiently:

| | |
|---|---|
| COUNT$(P, m)$: | Return the number of occurrences of $P$ in $T$. |
| LOCATE$(P, m)$: | Return all occurrences of pattern $P$ in $T$. |
| EXTRACT$(i, j)$: | Extract $T[i..j]$ from the self-index. |

Self-indexes typically provide this functionality by allowing the following basic operations over $T^{\text{BWT}}$:

| | |
|---|---|
| ACCESS$(T^{\text{BWT}}, i)$: | Return $T^{\text{BWT}}[i]$. |
| RANK$(T^{\text{BWT}}, i, c)$: | Return the number of times symbol $c$ occurs in $T^{\text{BWT}}[0..i-1]$. |
| SELECT$(T^{\text{BWT}}, i, c)$: | Return the position of the $i$-th occurrence of symbol $c$ in $T^{\text{BWT}}$. |

To support these operations, a wavelet tree [7] is built over $T^{\text{BWT}}$ which supports all operations in $\mathcal{O}(\log \sigma)$ time. Wavelet trees over $T^{\text{BWT}}$ take roughly the space of the compressed representation of $T$ [6]. For an overview of wavelet trees refer to Navarro [14]. The main component of all operations in self-indexes is *backward search*, where all rows in $\mathcal{M}$ prefixed by $P$ can be found in $\mathcal{O}(m \log \sigma)$ time by processing the pattern backwards by calculating LF() $2m$ times in $\mathcal{O}(\log \sigma)$ time as shown in Equation 1.

$$\text{LF}(i) = \text{LF}(i, c) = C[c] + \text{RANK}(T^{\text{BWT}}, i, c) \quad (1)$$

where $c$ is the symbol $T^{\text{BWT}}[i]$, and $C[c]$ stores the number of symbols in $T^{\text{BWT}}$ smaller than $c$. For a more detailed overview of self-indexes refer to Navarro and Mäkinen [16] or Ferragina et al. [6]. Similar techniques are used to allow searching in $T^{k\text{-BWT}}$ [21].

## 3. THE VARIABLE DEPTH TRANSFORM

The $k$-BWT sorts each suffix up to a fixed depth of $k$. Figure 1 shows the context size distribution for sorting depth $k = 2$ to 8. Note that as the sorting depth increases, the number of small contexts increase. However, even at a depth of 8, many large contexts groups remain which correspond to substrings in $T$ that have a length of 8. Instead of sorting all rows deeper, we sort only context groups above a threshold $v$.

The $k$-BWT specifies the sorting depth $k$ until the rows in $\mathcal{M}_k$ are sorted. The incomplete sorting of $\mathcal{M}_k$ results in rows in $\mathcal{M}_k$ being grouped together in context groups. Each row in an individual context group shares the same $k$ symbol prefix. Instead of defining the sorting depth $k$, we define the maximum context group size $v$ in $\mathcal{M}_v$ allowed. We continue to sort context groups with more than

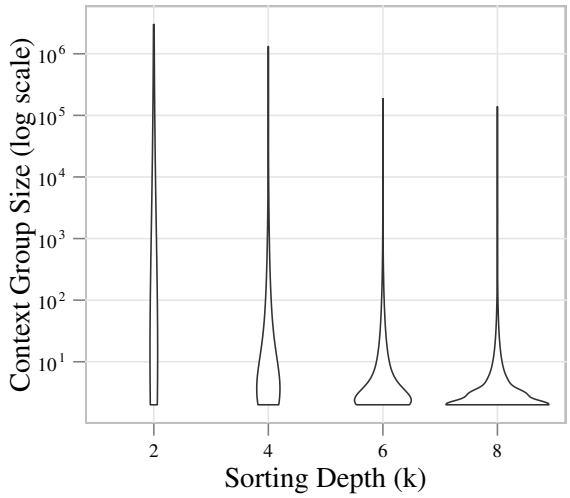



**Figure 1: Context group size (logarithmic) distribution for different sorting depth $k$ of the $k$-BWT for a 100 MB English text file.**

$v$ rows until the resulting context groups contain at most $v$ rows. This implies that different parts of $\mathcal{M}_v$ will be sorted to different depths as not all $k$-grams in $T$ occur equally often. Figure 2 shows an example of the $v$-BWT. Context groups '`p`' and '`$`' are sorted up to a depth 1. Context groups '`ay`' and '`yay`' are sorted to depth 2 and 3 respectively.

| i | $D_v$ | LF | **F** | | | | | | | | | | | $L$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 11 | **$** | y | a | y | a | y | a | p | y | a | y | a |
| 1 | 1 | 10 | **a** | **$** | y | a | y | a | y | a | p | y | a | y |
| 2 | 1 | 5 | **a** | **p** | y | a | y | a | $ | y | a | y | a | y |
| 3 | 1 | 1 | **a** | **y** | a | y | a | p | y | a | y | a | $ | y |
| 4 | 0 | 3 | **a** | **y** | a | p | y | a | y | a | $ | y | a | y |
| 5 | 0 | 8 | **a** | **y** | a | $ | y | a | y | a | y | a | p | y |
| 6 | 1 | 6 | **p** | y | a | y | a | $ | y | a | y | a | y | a |
| 7 | 1 | 9 | **y** | **a** | **$** | y | a | y | a | y | a | p | y | a |
| 8 | 1 | 4 | **y** | **a** | **p** | y | a | y | a | $ | y | a | y | a |
| 9 | 1 | 0 | **y** | **a** | **y** | a | y | a | p | y | a | y | a | $ |
| 10 | 0 | 2 | **y** | **a** | **y** | a | p | y | a | y | a | $ | y | a |
| 11 | 0 | 7 | **y** | **a** | **y** | a | $ | y | a | y | a | y | a | p |

**Figure 2: $v$-BWT for $T$ =`yayayapyaya$` including LF mapping and the context-group vector for threshold $s = 3$. The different sorting depths are boldfaced.**

The bitvector $D_v$ describing the context group boundaries is now defined as shown in Definition 2. $D_v[i]$ is 0 if $D_{v-1}[i]$ is 0 and the size of the context group containing row $i$ in the previous sorting stage ($d^i_{v-1}$) was smaller or equal to $v$ or if the $v$-prefix of row $i$ is equal to row $i-1$. $D_v[i]$ is 1 otherwise.

**Algorithm 1** $v$-BWT Forward Transform of text $T$ with threshold $v$

```
VBWT (T[0...n-1], v)
Initialize SA[0...n-1]
Count symbols to create B_1
for each context group D_1[i...j] do
    RADIXSORT (T, SA, D_1[i...j], v, 2)
end for
for i ← 0 to n-1 do
    if SA[i] = 0 then
        T^{v-BWT}[i] ← T[n-1]
    else
        T^{v-BWT}[i] ← T[SA[i]-1]
    end if
end for
return T^{v-BWT}
```

FUNCTION RADIXSORT $(T, \text{SA}, D[i \dots j], v, k)$

```
if j - i + 1 ≤ v or k ≥ k_max then
    return
end if
COUNTSORT symbols in SA[i...j]
for all symbols ∈ SA[i...j] do
    Mark start of new context group in D
    RADIXSORT (T, SA, D[i...j], v, k+1)
end for
```

DEFINITION 2. *For any $1 \leq v < n$, let $d^i_{v-1}$ be the size of the context group containing row $i$ after sorting step $v-1$. Let $D_v[0, n-1]$ be a bitvector, such that $D_v[0] = 1$ and, for $1 \leq i < n$,*

$$D_v[i] = \begin{cases} 0 & \text{if } D_{v-1}[i] = 0 \text{ and } d^i_{v-1} \leq s \\ 0 & \text{if } D_{v-1}[i] = 0 \text{ and } d^i_{v-1} > s \text{ and} \\ & \mathcal{M}_v[i][0, v-1] = \mathcal{M}_v[i-1][0, v-1] \\ 1 & \text{otherwise} \end{cases}$$

The forward transformation of the $v$-BWT is outlined in Algorithm 1. We recursively perform radixsort for each of the context groups until the context group size is less then our defined threshold $v$. The algorithm returns the context group vector $D_v$, as well as the suffix array (SA) sorted up to variable sorting depth. The duality between the BWT and suffix arrays is used to create $T^{v\text{-BWT}}$ as $T^{v\text{-BWT}}[i] = T[\text{SA}[i] - 1]$.

In order to bound worst-case sorting time, additional parameters are necessary. Let $k_{min}$ to be the minimum sorting depth for all context groups and let $k_{max}$ to be the maximum sorting depth. This guarantees a worst case runtime complexity of $\mathcal{O}(k_{max} n)$. In practice small values for the parameter $k$ allow the bounded sorting depth transform to perform faster than full suffix array construction algorithms [4].

## 3.1 Transform Reversal

The $k$-BWT can be reversed using the bit vector $D_k$ marking the beginning of the context boundaries as contexts are required to be processed in reverse sequential order. The LF() mapping is only guaranteed to "jump" into the correct context group [21]. Similarly, $D_v$ can be used to reverse the $v$-BWT even though not all columns are sorted to the same depth $k$.

LEMMA 1. *The text $T$ can be recovered from the permutation $T^{v\text{-BWT}}$ using the context group boundaries $D_v$ and the* LF() *mapping.*

**Proof.** Equation 1 counts the number of occurrences of $c = T^{\text{BWT}}[i]$ in $T^{\text{BWT}}[0, i]$. If $i$ represents the last row of a context group, the set of rows in $\mathcal{M}[0, i]$ is identical to the rows in $\mathcal{M}_v[0, i]$ as the context groups are lexicographically sorted. Therefore the number of occurrences of any $c$ in $T^{v\text{-BWT}}[0, i]$ is the same as in $T^{\text{BWT}}[0, i]$. The different sorting depths $k'$ and $k''$ do not affect the lexicographical order of different contexts as the sorting depth can only affect the order within a context group. So, $j = \text{LF}(i)$ maps correctly between two context groups $d^i_{k'}$ and $d^j_{k''}$ despite being sorted to different depths $k'$ and $k''$. As $\text{LF}(i)$ must map to the correct preceding context group as in the $k$-BWT, using $D_v$ we can process each context group in reverse sequential order to recover $T$. ■

To recover $T$ from $T^{v\text{-BWT}}$ no additional information is required as the context boundaries $D_v$ can be recovered from $T^{v\text{-BWT}}$ in $\mathcal{O}(k_{max}n)$ time, where $k_{max}$ is the maximum sorting depth of any context group in $T^{v\text{-BWT}}$.

LEMMA 2. *$D_k$ can be recovered directly from $T^{v\text{-BWT}}$ using no additional information.*

**Proof.** Recall that context information is not needed to restore the first $k$ columns of $\mathcal{M}_k$. Instead of recovering $\mathcal{M}_v$ to a depth of $k$, we can recover based on the number of rows, $s$, with an identical prefix $\mathcal{M}_v[1...j]$. Let $t$ be the the maximum number of rows in $\mathcal{M}_v$ that have the same prefix $\mathcal{M}_v[1...j]$ when sorted to a depth of $j$. If the number of rows $t$ with the same prefix exceeds $s$ at the current sorting depth $j$, this context group must be sorted up to depth $j + 1$. We continue recovering $\mathcal{M}_v$ in the current context group until $t \leq s$. The final context group recovered is $D_v$. ■

We now give an example of how to recover $D_v$ from $T^{v\text{-BWT}}$. First, we recover $F$ by sorting $L = T^{v\text{-BWT}}$ and initialize $D_1$ to the symbol boundaries. We also keep track of the $F \rightarrow L$ column mapping:

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $D_1$ | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| $F$ | \$ | a | a | a | a | a | p | y | y | y | y | y |
| $L$ | a | y | y | y | y | y | a | a | a | \$ | a | p |
| $FL_1$ | 9 | 0 | 6 | 7 | 8 | 10 | 11 | 1 | 2 | 3 | 4 | 5 |

Next, for all context groups larger or equal $s = 3$, we recover the next column in $\mathcal{M}$ using the initial $FL_1$ mapping. We update $D_2$ to include the new context boundaries and use the initial $FL_1$ mapping to create $FL_2[i] = FL_1[FL_1[i]]$ for context groups larger than $v$.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $D_2$ | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| $F$ | \$ | a | a | a | a | a | p | y | y | y | y | y |
| | | \$ | p | y | y | y | | a | a | a | a | a |
| $L$ | a | y | y | y | y | y | a | a | a | \$ | a | p |
| $FL_2$ | | | | | | | | 0 | 6 | 7 | 8 | 10 |

Using $FL_2$ we recover the next column for context groups larger than $v$ in a similar manner:

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $D_3$ | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| $F$ | \$ | a | a | a | a | a | p | y | y | y | y | y |
| | | \$ | p | y | y | y | | a | a | a | a | a |
| | | | | | | | | \$ | p | y | y | y |
| $L$ | a | y | y | y | y | y | a | a | a | \$ | a | p |

We now have $D_v$ as the size of all of the context groups less than or equal to $v$, and can therefore be used to recover $T$ from $T^{v\text{-BWT}}$.

# 4. VARIABLE LENGTH Q-GRAM INDEX

A $q$-gram is a contiguous sequence of symbols in a text $T$: $T[i..\ell]$. A $q$-gram index uses all $q$-grams in $T$ to support approximate pattern matching over the text [15]. Traditional $q$-gram indexes are based on inverted files. For each distinct $q$-gram $q_i$ in $T$, a list of positions of all occurrences of $q_i$ are stored. These list can be $d$-gapped and compressed to reduce space. Individual inverted files are accessed through the vocabulary, which can be represented using a data structure such as a trie [18]. In large text collections, $q$-gram indexes have a few serious limitations. First, the number of distinct $q$-grams in $T$ can grow exponentially with the size of $q$ in the worst case. Second, certain $q$-grams tend to occur much more frequently than others.

Navarro and Salmela [18] propose a variable length $q$-gram index, where each variable length $q$-gram is required to have a uniform number of occurrences, and no $q$-gram occurs more than $s$ times. The index is prefix-free, so no "selected" $q$-gram can be a prefix of any other $q$-gram in the index. To create the index, Navarro and Salmela first construct a suffix tree over $T$ in $\mathcal{O}(n)$ time. Next the suffix tree is traversed in depth first order in $\mathcal{O}(n)$ time to retrieve the vocabulary of the index by pruning the suffix tree at nodes whose subtree contains at most $s$ leaf nodes corresponding to suffix positions in $T$. Next, the position lists are sorted in increasing order in $\mathcal{O}(n \log \sigma)$ time and compressed in $\mathcal{O}(n)$ time. The total cost of constructing the index is therefore $\mathcal{O}(n \log \sigma + n \log s)$.

The $v$-BWT can significantly simplify the construction of a variable $q$-gram index. First, we create $T^{v\text{-BWT}}$ of $T$ with threshold $v$. In the process the following components of the $q$-gram index can be created. The suffix tree partitioning of Navarro and Salmela [18] can be accomplished using $D_v$ since each context group contains at most $v$ rows. The postings lists can be obtained implicitly via $\text{SA}_v$, the suffix array used to sort $T$. Within each context group, the suffix array positions correspond to the entries in the postings list in the $q$-gram index. These lists are already sorted and do not require the $\mathcal{O}(n \log \sigma)$ sort described by Navarro and Salmela. In fact, we perform this step implicitly *while* creating the partitioning.

## 4.1 Representing the Vocabulary

Traditional $q$-gram indexes consist of two main components. The vocabulary stored as a trie, and a compressed inverted file for each distinct indexed $q$-gram containing all occurrences of the $q$-gram in $T$. To perform an approximate pattern search, a pattern is split up into $k + 1$ substrings. Next, for each substring the inverted list is loaded by querying the vocabulary. Previously we showed how to obtain a variable $q$-gram partitioning using the $v$-BWT. Here we show how we can replace the vocabulary of a variable $q$-gram index with a wavelet tree over $T^{v\text{-BWT}}$.

The $v$-BWT can be used to obtain a variable length $q$-gram partitioning equivalent to the index proposed by Navarro and Salmela [18]. Instead of using a trie to store the vocabulary, we can instead perform a backwards search using a compressed wavelet tree over $T^{v\text{-BWT}}$.

LEMMA 3. *Backwards search for any substring $p_i$ can be performed in $T^{v\text{-BWT}}$ as long as the number of matching rows, $[sp, ep]$ in $\mathcal{M}_v$ are $\geq v$.*

**Proof.** Petri et al. [21] show that performing backwards search for a pattern up to length $k$ works correctly in $T^{k\text{-BWT}}$ as each context is guaranteed to be sorted up to depth $k$. Therefore, performing $k - 1$ backwards probes is guaranteed to return the correct range of rows, $sp, ep$, in $\mathcal{M}_k$ for any $p_i$ of length $k$. Similarly, every context group

$d_k^i$ corresponding to a prefix $\mathcal{M}_v[0..j]$ is sorted if there are more than $v$ rows in $\mathcal{M}_v$ prefixed by $\mathcal{M}_v[0..j]$ in $T^{v\text{-BWT}}$. Therefore, backwards search is guaranteed to result in the correct $sp, ep$ in $\mathcal{M}_v$ if $ep - sp + 1 \geq v$. ■

So, we use a wavelet tree over $T^{v\text{-BWT}}$ to determine ranges in $\mathcal{M}_v$ which correspond to substrings $p_i$ of $P$. The size of the range corresponds to the number of occurrences of $p_i$ in $T$. For patterns with less than $v$ occurrences, the range in $\mathcal{M}_v$ is not guaranteed to be continuous, so the $i-1$ context must be used instead. When this happens, all occurrences are still found, but the number of verifications is not guaranteed to be minimal.

## 4.2 Optimal Pattern Partitioning

To search for a pattern $P$ with at most $k$ errors, a $q$-gram index performs a *filtering* step whereby a string $A$ is split into $k+1$ substrings $a_1 \dots a_{k+1}$. For $A$ to occur in a string $B$ with at most $k$ errors, at least one substring $a_i$ must appear in $B$ [19]. A $q$-gram index is used to find all *candidate* positions of $P$ in $T$ by partitioning $P$ into $k+1$ substrings $p_1 \dots p_{k+1}$ and retrieving the positions in $T$ for all $p_i$. Navarro and Baeza-Yates [15] provide a dynamic programming algorithm which calculates the optimal partitioning of $P$ into $k+1$ pieces to minimize the number of candidates. In the second step, a standard *edit distance* algorithm is then used to verify all candidates [17].

We now show how to use the optimal pattern partitioning algorithm proposed by Navarro and Baeza-Yates [15] and later used by Navarro and Salmela [18] to enable approximate searching using a wavelet tree over $T^{v\text{-BWT}}$. The key intuition of Navarro and Baeza-Yates's algorithm is to compute all $m^2$ possible substrings $P[i-j]$ and the resulting candidate list lengths in a matrix $R[i,j]$ of size $\mathcal{O}(m^2)$. Dynamic programming is then used to to retrieve the optimal partitioning by processing $R$ in $\mathcal{O}(m^2 k)$ time [15, 18]. Using the backwards search (`BWS`) procedure, we compute $R[i,j]$:

$$R[i,j] = \begin{cases} |\langle sp, ep\rangle| & \text{if } \texttt{BWS}(P[i-j]) = |\langle sp, ep\rangle| \geq v \\ \infty & \text{otherwise} \end{cases}$$

Where $\langle sp, ep\rangle$ is the range in the suffix array prefixed by $P$. This range is only guaranteed to be continuous if $|\langle sp, ep\rangle| \geq v$, as within a context group rows are not lexicographically sorted. All substrings for which we cannot determine $\langle sp, ep\rangle$ are set to infinity in our calculations, thus making sure they are not included in the final partitioning of $P$ into $p_{ij}, ..p_{j+1,l}, ...p_{m-1}$. For each substring we retrieve the corresponding $\langle sp, ep\rangle$ ranges in order to determine the parts of the suffix array containing the candidate positions.

## 4.3 Storing Postings Lists

Traditionally, the vocabulary contains pointers (file offsets) at which the individual postings list for the indexed strings ($q$-grams) are stored. As we are using a wavelet tree to store the vocabulary, we choose a different representation to store postings lists. Recall that within a context group in $T^{v\text{-BWT}}$, all corresponding suffix array positions are in ascending text order. We can therefore store a compressed version $\text{SA}'_v$ of $\text{SA}_v$ which $d$-gaps and compresses all offsets in a single context group in the same manner as is often used in postings lists for inverted indexes.

Unfortunately, the ranges $\langle sp, ep\rangle$ in $\text{SA}_v$ cannot be used to find the corresponding position in $\text{SA}'_v$. So, we store an additional bitvector $D'_v$ that maps context groups in $\text{SA}_v$ to the corresponding starting positions in $\text{SA}'_v$. First we calculate the distance of $sp$ to the corresponding context group start in $\text{SA}_v$ using $\ell = \text{RANK}(D_v, sp, 1)$ and $t = \text{SELECT}(D_v, \ell, 1)$. Next we map the context group into the compressed representation $\text{SA}'_v$ using $sp' = \text{SELECT}(D'_v, t, 1)$. Starting from $sp'$ we skip the first $sp - \ell$ encoded numbers and then retrieve the next $ep - sp + 1$ encoded positions of $\langle sp, ep\rangle$. Note that $\langle sp, ep\rangle$ might span multiple smaller context groups which must each contain separately compressed $d$-gap lists.

The vocabulary and all auxiliary information needed to perform optimal partitioning can be stored using $H_{k_{min}}(T)$ space – the cost of storing a wavelet tree over $T^{v\text{-BWT}}$ with a minimal sorting depth of $k_{min}$. The text positions in $\text{SA}'_v$ use variable byte coding which uses up to 30% more space than bit-compressed inverted lists, but allows for faster decoding time [25]. We further store $H_0$ compressed representations of $D_v$ and $D'_v$ [22].

# 5. EXPERIMENTS

## 5.1 Experimental Setup

In our experiments we use two datasets. We use the first 1 GB of a genome sequence created by concatenating the "Soft-masked" assembly sequence of the human genome (hg19/GRCH37) and the Dec. 2008 assembly of the cat genome (catChrV17e) in FASTA format. We remove all comment/section separators and replaced them with a separator token to fix the alphabet size. We call this data set DNA. Our second data set was generated from from the 2009 Clueweb web crawl available at `http://lemurproject.org/clueweb09.php/`. The first 64 WARC files in the directory `Clueweb09/disk1/Clueweb09_English_1/enwp00/` were concatenated together and null bytes in the text were replaced with 0xFF-bytes. The first 1 GB were used in our experiments which we denote as WEB.

We use a server with 2× Intel Xeon E5640 Processors with a 12 MB L3 cache, 144 GB of DDR3 DRAM running Ubuntu Linux version 12.04. The `g++` compiler version 4.6.3 with the basic compile option `-O3 -DNDEBUG -funroll-loop` was used. For basic succinct data structures, we use the succinct data structure library (`sdsl`) available at `http://github.com/simongog/sdsl/`. For suffix array construction we use the `libdivsufsort` library available at `http://code.google.com/p/libdivsufsort/`. In our $v$-BWT transform implementation, we used the cache efficient radixsort implementation proposed by Kärkkäinen and Rantala [9]. To compare our wavelet tree based vocabulary, we use a Hu-Tucker front-coding based vocabulary proposed by Brisaboa et al. [1].

## 5.2 Forward Transform Performance

Now we evaluate the runtime efficiency of our new transform and compare the forward transform with the $k$-BWT and the full BWT. We use the the suffix sorting algorithm implemented in `libdivsufsort` to construct the full BWT efficiently. Table 1 shows the runtime performance to create $T^{v\text{-BWT}}$, $T^{k\text{-BWT}}$ and $T^{\text{BWT}}$ respectively for both test files.

| | Time [sec] | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | $k$-BWT | | | $v$-BWT | | | | BWT |
| | 3 | 5 | 9 | 5 | 50 | 500 | 5000 | |
| DNA | 63 | 121 | 253 | 289 | 224 | 191 | 138 | 283 |
| WEB | 89 | 145 | 258 | 312 | 262 | 235 | 209 | 213 |

**Table 1: Construction time (in seconds) of $v$-BWT, $k$-BWT, and the full BWT using `divsufsort`**

The bounded transforms perform better for DNA than for WEB compared to the full BWT. For DNA, constructing the $k$-BWT is faster than constructing the full BWT. The $v$-BWT can be constructed more efficiently for sorting depths up to 5. Note that for $v = 5$, the $v$-BWT is "almost" identical to the full BWT, and only contexts up to size 5 remain. For $v = 50$ to 5000 the $v$-BWT can be constructed even more efficiently. The WEB data set can be constructed $40\%$ faster with the full BWT compared to DNA. Induced suffix sorting reduces the number of suffix comparisons required to construct the suffix array. Therefore, the number of suffix comparisons needed is the limiting factor. Longer text comparisons have to be performed to determine the order of two suffix positions. The bounded transforms also perform slower for WEB. For $v = 5$, the variable transform is $40\%$ slower than the induces suffix sorting method. As the sorting depth decreases, the $v$-BWT again outperforms the full BWT.

Overall the $v$-BWT can be constructed efficiently. However, we have not attempted to apply induced suffix sorting techniques commonly used during suffix array construction to speed up the construction process. This could potentially speed up the construction process significantly but remains future work.

## 5.3 Variable q-gram Index Construction

We now compare the construction time of our variable $q$-gram based index to the suffix tree method of Navarro and Salmela [18]. As described by Navarro and Salmela, we first construct a compressed suffix tree using `libsdsl`. Next we perform a depth first search traversal to determine the highest nodes in the suffix tree that have at most $v = 50$ children. The ranges in the suffix array corresponding to the marked nodes are recorded. Lastly, the individual ranges in the suffix array are sorted. We compare this approach to constructing an equivalent index using our $v$-BWT for $v = 50$. The different steps required in addition to the time required to build an index for threshold $v = 50$ for DNA are shown in Table 2. Note that the table further lists the cost to construct the vocabulary and compress the individual postings lists.

| | Time [sec] | |
|---|---|---|
| Step | suffix tree | $v$-BWT |
| construct CST | 736 | - |
| suffix tree traversal | 405 | - |
| sort suffix array | 453 | - |
| create $v$-BWT | - | 224 |
| build vocabulary | 24 | |
| vbyte compress postings lists | 30 | |
| Total | 1648 | 278 |

**Table 2: Construction cost comparison of the method by Navarro and Salmela and the $v$-BWT for $v = 50$ on the DNA data set.**

As expected, the construction of the suffix tree is the main bottleneck in the method of Navarro and Salmela. In fact, traversing the suffix tree to determine the different ranges in the suffix array for the approach is more expensive than creating the entire index using the $v$-BWT transform. Sorting each range in the suffix array in the suffix tree method is also computationally expensive, and unnecessary when using $v$-BWT. Overall, the $v$-BWT index can be constructed 5 times faster than the best known $v$-gram approach.

## 5.4 Variable q-gram Verifications

All $q$-gram based approximate pattern matching approaches use filtering to reduce verification costs. Potential matching candidates must still be verified using an *edit distance* algorithm. The goal of the filter is to minimize the number of verifications required to perform approximate search. We now evaluate the number of verifications required by each indexing approach. First, we perform 1000 approximate pattern searches for pattern lengths 20 to 50 using different error levels. The patterns were randomly sampled from each data set. Figure 3 shows the number of candidate positions which must be verified after pattern partitioning is performed. For this experiment, we only compare $k = 5$ and $v = 50$ using a wavelet tree as the vocabulary for both approaches. For DNA, the number of positions requiring verification tend to be higher than for WEB as the data is more uniform, and the alphabet size is smaller. The $v$-BWT always outperforms classical $k$-BWT partitioning. The variance in the WEB data set is higher than for DNA, while DNA generally requires more verifications using the $k$-BWT based approach. The $v$-BWT approach outperforms the $k$-BWT approach for the DNA data set by several orders of magnitude except for patterns of length 20 with error rates of 3 and 4. This implies that $P$ has to be split into 4 and 5 substrings respectively. As the sorting depth for the $k$-BWT is 5, we conjecture that the substrings being evaluated with the $v$-BWT are rarely longer than in the $k$-BWT.

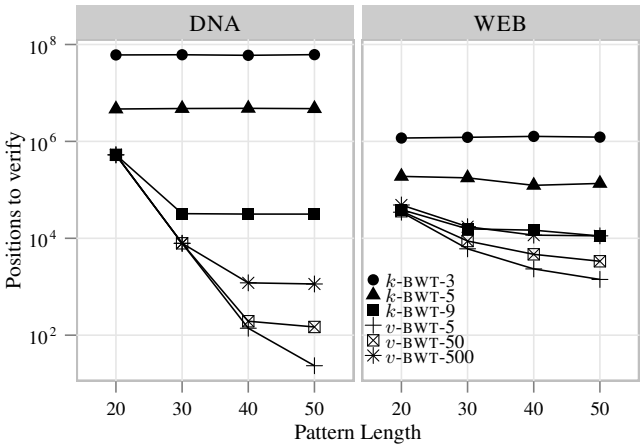



**Figure 4: Number of verifications required for $k$-BWT for variable $k = 3, 5, 9$ and $v = 5, 50, 500$ for 2 errors for DNA and WEB data sets.**

Next, we show how the number of verifications varies with different sorting parameters. We choose only small $k$ values as the number of potential dictionary entries can, in the worst case, grow exponentially as $k$ increases. Similarly, we choose the parameter $v$ to have similar construction costs as our chosen $k$ values. Figure 4 shows the *mean* number of verifications required for 1000 approximate pattern searches for patterns of length 20 to 50 for variable transform parameters. The number of verifications required using the standard fixed $q$-gram $k$-BWT approach decreases as $k$ increases due to the fact that longer substrings can be matched. For $k = 9$, performance is similar to that of the $v$-BWT for patterns of length 20. Generally, for all $k$ the $k$-BWT approach requires more verifications. The average number of verifications required stays roughly constant for the fixed $q$-gram approach whereas the mean number of verifications decreases using the variable length $q$-gram approach as the length of the pattern increases. As the pattern length increases, our approach can match longer variable length $q$-grams during the optimal partitioning phase. Longer $q$-grams occur less frequently. Therefore, the number of verifications required decreases.

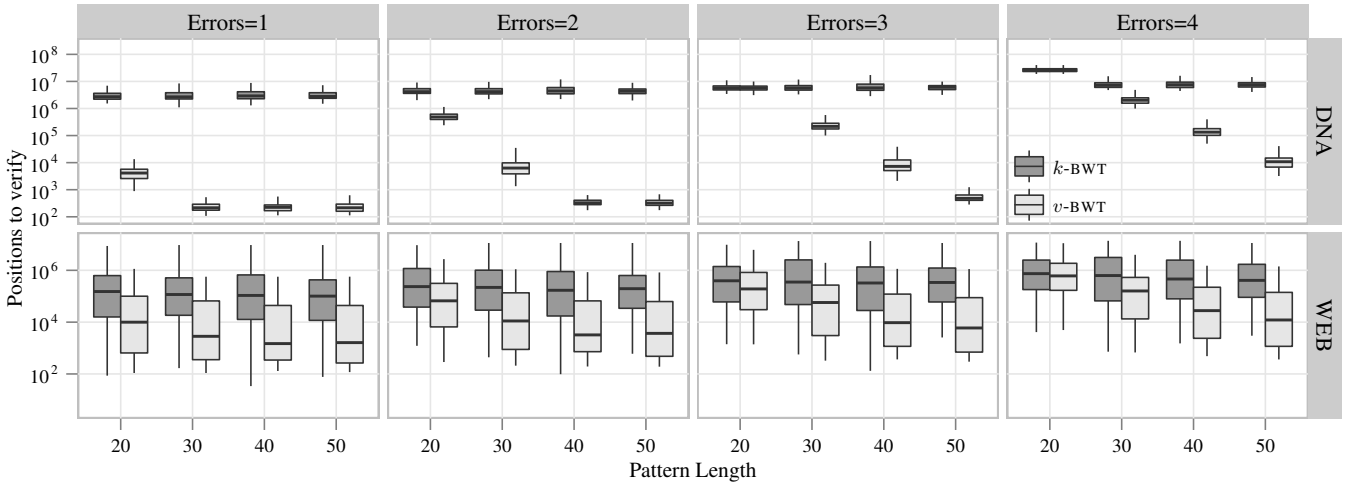



**Figure 3: Number of verifications required for $k$-BWT with $k = 5$ and $v$-BWT with $v = 50$ for the DNA and WEB data sets.**

## 5.5 Variable q-gram Vocabularies

Last, we evaluate the performance of a wavelet tree based vocabulary by comparing the performance to a Hu-Tucker front coding (HTFC) vocabulary as proposed by Brisaboa et al. [1]. We compare the running time required by each vocabulary type to execute the optimal partitioning algorithm with the space required to store the vocabulary.

We use the HTFC dictionary as follows: Insert the first row of each context group in $M_k$ and $M_v$ into the HTFC vocabulary. For the $k$-BWT based approach, insert the suffix with length $k$. For the $v$-BWT, insert the suffix that maximizes the longest common prefix (LCP) of the adjacent context groups. For example, given the context group corresponding to `bba` with adjacent contexts groups `ba` and `bbbb`, insert `bba`. Since the sorting depth for each context group is not stored, the unique prefix representing the context group is calculated during insertion into the vocabulary. Inside the HTFC vocabulary, each string is inserted into a block of size $B$ and compressed using front coding and Hu-Tucker coding. During the query phase, binary search is performed over the first entries of each block to find the candidate block the search string must occur in. Next, the block is sequentially decompressed until the string is found, or the next uncompressed block entry is larger than the search string. The overall performance of the vocabulary depends on the block size $B$ (which determines the number of sequential decompression steps), and the number of blocks in the vocabulary (which depends on $B$ and the number of strings inserted). We modify the original HTFC of Brisaboa et al. [1] to support prefix search by returning the first and last entry in the vocabulary for which a given search string is a prefix. On a successful search for a pattern $P$, the vocabulary returns to numbers $i, j$ corresponding to the positions of the first and last entry in the vocabulary prefixed by $P$. We then use $D_v$, the bitvector describing the context group boundaries, to calculate the number of times $P$ occurs in $T$ by determining the $i$-th and $j$-th one bit in $D_v$: $sp = \text{SELECT}(D_v, i, 1)$ and $ep = \text{SELECT}(D_v, j + 1, 1) - 1$. We can then determine the number of occurrences of $P$ in $T$ as $occ = ep - sp + 1$.

For the wavelet tree vocabulary we use three different Huffman shaped wavelet trees. The first wavelet tree uses uncompressed bitvectors (`wt-bv`). The second (`wt-15`) and third (`wt-63`) use $H_0$ compressed vectors proposed by Raman et al. [22]. By using compressed bitvectors and a Huffman shaped wavelet tree the cost of storing $T^{v\text{-BWT}}$ is roughly equal to the size of the compressed representation of $T$. We determine the number of occurrences of a pattern $P$ by performing backwards search as described previously. Figure 5 shows the time-space trade-offs for DNA and the $v$-BWT with $v = 5, 50, 500$. We show the mean time in seconds per partitioning step compared to the vocabulary size in MB.

For $v = 500$ the HTFC vocabulary outperforms both wavelet tree dictionaries using compressed bitvectors (`wt-15` and `wt-63`) while the wavelet tree using uncompressed bitvectors is much faster, but also uses much more space. This can be explained as follows. For $v = 500$, the number of context groups is small. Therefore, not many strings are inserted into the HTFC while the wavelet tree always contains all rows in $M_v$. Searching in the HTFC vocabulary depends on the number of strings in the vocabulary. Therefore, for $v = 500$ with block sizes $B = 5, 50$, the HTFC vocabulary is both smaller and roughly as fast as `wt-15` and `wt-63`.

However, as the sorting requirements of each context group is increased, the wavelet tree becomes more competitive in space usage. For $v = 50$, more strings are inserted into the HTFC vocabulary, requiring more space. Still, the compressed wavelet trees are roughly twice as large as the HTFC based vocabulary. When $v = 5$, the space required for the HTFC vocabulary is larger than the wavelet tree. The wavelet tree using uncompressed bitvectors now uses less space while allowing much faster search. As the sorting depth is increased, the wavelet tree approach becomes more compelling. Moreover, increasing the sorting depth also decreases the number of verifications required. Therefore, using a wavelet tree based vocabulary can be both space efficient and support efficient filtering-based approximate pattern matching.

# 6. CONCLUSION AND FUTURE WORK

We have presented a new context based sort transformation: the $v$-BWT. We show how the transform differs from previous context sorting transforms. In addition, we show that the $v$-BWT can be used to create text indexes which can be used for approximate pattern matching. Our experimental evaluation shows that the transform can be used to construct variable length $q$-gram indexes five times faster than previous methods. We show that the number of verifications that have to be performed using a variable $q$-gram index are less than traditional fixed $q$-gram based indexes. We further show that using wavelet trees over the transform output can be used as the vocabulary component in the approximate index.

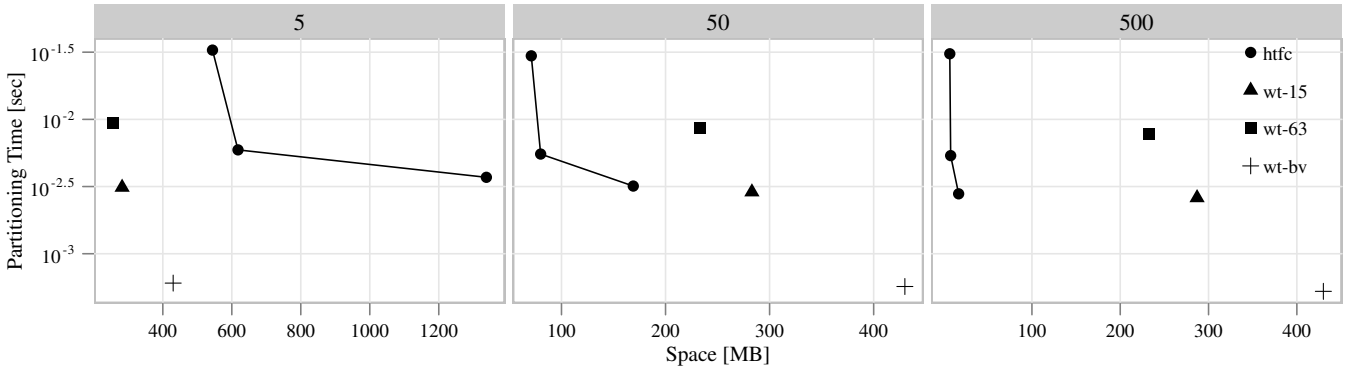



**Figure 5: Time and Space trade-offs during optimal pattern partitioning for HTFC ($B = 5, 50, 500$) and wavelet tree based dictionaries for DNA using $v$-BWT and $v = 5, 50, 500$.**

Future work includes: Adopting fast, induced suffix sorting based BWT construction algorithms to construct $T^{v\text{-BWT}}$; Using the transform to construct suffix arrays on disk; and exploring the viability of variable length q-gram indexes for a wide variety of common IR and Bioinformatics search problems.

*Acknowledgement.* This work was supported in part by the Australian Research Council and by NICTA. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

## References

[1] N. R. Brisaboa, R. Cánovas, F. Claude, M. A. Martínez-Prieto, and G. Navarro. Compressed string dictionaries. In *SEA*, pages 136–147, 2011.

[2] M. Burrows and D. J. Wheeler. A block-sorting lossless data compression algorithm. Technical Report 124, Digital Equipment Corporation, Palo Alto, California, May 1994.

[3] H. L. Chan, T. W. Lam, W. K. Sung, S. L. Tam, and S. S. Wong. Compressed indexes for approximate string matching. *Algorithmica*, 58(2):263–281, 2010.

[4] J. S. Culpepper, M. Petri, and S. J. Puglisi. Revisiting bounded context block-sorting transformations. *Software Practice and Experience*, 42 (8):1037–1054, August 2012.

[5] P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *FOCS*, pages 390–398, 2000.

[6] P. Ferragina, R. González, G. Navarro, and R. Venturini. Compressed text indexes: from theory to practice. *Journal of Experimental Algorithmics*, 13:1.12–1.31, 2009.

[7] R. Grossi, A. Gupta, and J. S. Vitter. High-order entropy-compressed text indexes. In *SODA*, pages 841–850, 2003.

[8] D. Gusfield. *Algorithms on Strings, Trees, and Sequences.* Cambridge University Press, New York, New York, USA, 1997.

[9] J. Kärkkäinen and T. Rantala. Engineering radix sort for strings. In *SPIRE*, pages 3–14, 2008.

[10] K. Kukich. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24(4):377–439, 1992.

[11] H. Li and R. Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[12] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *SIGIR*, pages 472–479, 2005.

[13] G. Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, 2001.

[14] G. Navarro. Wavelet trees for all. In *CPM*, pages 2–26, 2012.

[15] G. Navarro and R. A. Baeza-Yates. A practical $q$-gram index for text retrieval allowing errors. *CLEI Electron. J.*, 1(2), 1998.

[16] G. Navarro and V. Mäkinen. Compressed full-text indexes. *ACM Comput. Surv.*, 39(1), 2007.

[17] G. Navarro and M. Raffinot. *Flexible Pattern Matching in Strings – Practical on-line search algorithms for texts and biological sequences*. Cambridge University Press, 2002. ISBN 0-521-81307-7. 280 pages.

[18] G. Navarro and L. Salmela. Indexing variable length substrings for exact and approximate matching. In *SPIRE*, pages 214–221, 2009.

[19] G. Navarro, R. A. Baeza-Yates, E. Sutinen, and J. Tarhio. Indexing methods for approximate string matching. *IEEE Data Eng. Bull.*, 24 (4):19–27, 2001.

[20] G. Nong, S. Zhang, and W. H. Chan. Computing inverse ST in linear complexity. In *CPM*, pages 178–190, 2008.

[21] M. Petri, G. Navarro, J. S. Culpepper, and S. J. Puglisi. Backwards search in context bound text transformations. In *CCP*, pages 82–91, 2011.

[22] R. Raman, V. Raman, and S. S. Rao. Succinct indexable dictionaries with applications to encoding k-ary trees and multisets. In *SODA*, pages 233–242, 2002.

[23] L. Russo, G. Navarro, A. Oliveira, and P. Morales. Approximate string matching with compressed indexes. *Algorithms*, 2(3):1105–1136, 2009.

[24] M. Schindler. A fast block-sorting algorithm for lossless data compression. In J. A. Storer and M. Cohn, editors, *DCC*, page 469, Los Alamitos, California, March 1997. IEEE Computer Society Press.

[25] F. Scholer, H. E. Williams, J. Yiannis, and J. Zobel. Compression of inverted indexes for fast query evaluation. In *SIGIR*, pages 222–229, 2002.

[26] S. Wu and U. Manber. Fast text searching allowing errors. *Communications of the ACM*, 35(10):83–91, 1992.

[27] H. Yokoo. Notes on block-sorting data compression. *Electronics and Communications in Japan, Part 3*, 82(6):18–25, 1999.

# An Attempt to Measure the Quality of Questions in Question Time of the Australian Federal Parliament

Andrew Turpin
University of Melbourne
aturpin@unimelb.edu.au

## ABSTRACT

This paper uses standard information retrieval techniques to measure the quality of information exchange during Question Time in the Australian Federal Parliament's House of Representatives from 1998 to 2012. A search engine is used to index all answers to questions, and then runs each question as a query, recording the rank of the actual answer in the returned list of documents. Using this rank as a measure of quality, Question Time has deteriorated over the last decade. The main deterioration has been in information exchange in "Dorothy Dixer" questions. The corpus used for this study is available from the author's web page for further investigations.

## Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Miscellaneous

## General Terms

Application

## Keywords

Information retrieval, Question Time, Parliament, Search Engine

## 1. INTRODUCTION

Question Time in the House of Representatives in the Australian Federal Parliament begins at 2pm on every sitting day, and is a period of about one hour where members of the parliament can ask Ministers questions and receive an answer typically limited to several minutes in length. While a seemingly essential part of a functioning democracy, Question Time has come under attack in recent days [10], and after the last election [1], for lacking content and relevance. Mark Rodrigues' publication on parliamentary reform from 2010 sums it up as follows.

> Question Time in the House of Representatives is often criticised for declining parliamentary standards and accountability. Oppositions are inclined to use partisan attacks disguised as questions to embarrass the government to which Ministers respond with lengthy answers of marginal relevance. Ministers often use Question Time to attack the opposition with pre-prepared statements in response to "Dorothy Dix" questions from their own side. Much of the theatre of Question Time is characterised by disorder contrived to make the evening news [9].

In this paper we use an off-the-shelf, open source information retrieval (IR) system to quantify whether questions and answers in Question Time contain less information and relevance in recent years than in past years. IR systems index a collection of documents, and then for any provided query, rank the documents in the collection against the query, with the top ranked document being deemed the most relevant to the query. In this study, all answers to all questions from Question Time in the period 1998 through to 2012 were indexed as a single document collection, then each question was run against the collection as a query. In theory, the answer provided by a parliamentarian to a question should be the top ranked document returned by the system; assuming, of course, that the question was answerable, and that they actually provided the answer. The hypothesis tested by this paper, therefore, is that the rank of answer documents to questions posed in 2012 are lower than the rank of answer documents in 1998. That is, questions and answers in 1998 contained more content that was decipherable by a search engine than 2012.

Hansard is a written daily record of everything that occurs in the parliament. The next section discusses the creation of the Question Time corpus from downloaded Hansard XML files: a non-trivial task, as it turned out. Then we present some results to address the hypothesis, and the final section of the paper discusses the results and limitations of this study.

## 2. CORPUS CREATION

All of the Hansard from 1998 to the current day can be downloaded from the Australian Parliament House Hansard Web site [4] in XML format. Unfortunately, from a computational point of view, the XML format changed mid-2011, and so different parsers are required to extract information for Hansards after March 2011 and before. This section describes how we extracted questions and answers from the


*ADCS '12* December 05 - 06 2012, Dunedin, New Zealand

XML. The final collection is available for download in TREC format from the author's home page.

The extraction begins with an XSL script to extract the "Questions Without Notice" `<debate>` from each Hansard file – one per day of sitting – and then the first `<question>` and `<answer>` from each `<subdebate.1>` of that debate. Sometimes each subdebate contains multiple questions and answers (called supplementary questions, in parliamentary language), but these were excluded from extraction as they are difficult to automatically validate. Often the supplementary questions are ruled "out of order" and so no answer is supplied, or the XML is incorrectly formed so that supplementary questions and answers are difficult to align. Furthermore, supplementary questions often contain references back to the original question (for example, pronouns), and so do not stand alone as suitable queries to an automated retrieval system.

Once each `<question>` and `<answer>` are identified, extracting the actual text is complicated by *interjections*. These are interruptions by people other than the nominated talker, and can occur from anywhere in the parliamentary chamber, including from The Speaker (chairperson of the debate). Post March 2011, these interjections are marked as separate paragraph tags that contain a `<span>` tag with an attribute that contains the string "Interjecting". For example,

```
<p class="HPS-Normal">
     <span class="HPS-Normal">
        <a href="EOH" type="MemberInterjecting">
        <span class="HPS-MemberInterjecting">
            Mr Laming:
        </span>
        </a>
            Superclinics?
     </span>
</p>
```

Thus all paragraphs are extracted within a question or answer that do not contain a `<span>` with an attribute that contains "Interjecting".

A second form of interruption to the text can come form The Speaker, either to call for order, or to address specific questions about the procedure of the debate ("Points of Order"). In these cases, the `<span>` tag contains an attribute that contains the substring "Office". For example,

```
<p class="HPS-Normal">
  <span class="HPS-Normal">
    <span class="HPS-OfficeContinuation">
       The DEPUTY SPEAKER:
    </span>
    The member for Wannon has got one last chance.
  </span>
</p>
```

Again, these paragraphs are excluded from the question and answer. The exact XSL used is given in Appendix A.

Ideally all sections of the debate would be tagged correctly, but often many people are speaking at once, and the debate moves at a furious pace. In these circumstances it is inevitable that mistakes are made, with interjections falling into parts of speeches, interruptions by The Speaker being tagged as part of a speech, and so on. Any errors that we came across during development of the XSL were corrected in the original XML by hand. In total we made 24 edits to files in 2012 and 1999. If those years are representative, then there are most likely 5 to 10 errors in the other years that remained uncorrected, and so may have filtered through to the final corpus.

In an effort to capture and correct errors in the final extracted question-answer pairs, and to tidy up the text for indexing, some post-processing of the XSL output was performed as described in Appendix A.

The raw XML contains other information that might be useful for analysis of Question Time, such as time stamps on speeches, names of speakers and interjectors, supplementary questions, and so on. For this study, the other piece of information extracted was the `<party>` of the questioner and responder. This allowed identification of "Dorothy Dixers": questions directed to members of the same political party to give them a platform for politicking. Occasionally a backbencher might ask a genuine question of their own front bench, but this was not distinguished in my analysis. We categorised the `<party>` fields into six parties as follows.

1. `AG`
2. `ALP`
3. `AUS`
4. `IND Ind Ind.`
5. `LP NATS NP Nats NaysWA NPActing CLP`
6. `UNKNOWN N/A`

Thus if the party of the questioner and responder fell into one of these categories, the question was deemed a Dorothy Dixer.

The final corpus contained 16310 question-answer pairs as outlined in Table 1 from 53 periods of parliament covering 941 sitting days.

The mean length of answers has remained about the same over all years: approximately 430 words or 190 words with stop words removed (linear regression, $R^2$ $p = 0.29$). The mean length of questions has decreased over time ($R^2 = 0.69, p = 0.0001$).

## 3. RANK OF ANSWERS

The most obvious experiment is simply to run each question as query and record the rank of the answer. For this purpose, we used the Zettair search engine (version 0.9.3) [12] with default Okapi BM25 [7, 8] parameters, stemming and stop list. The index created over all answers contained 16,310 documents, 54,971 distinct index terms, and 6,929,465 terms.

Figure 1 shows the median and inter-quartile ranges (IQR) of the ranks of the answer documents separated by year. The line indicates a simple linear regression. As can be seen, the median rank of the answer documents has significantly increased over the period of the corpus (t-test on slope, $p = 0.001$; $R^2 = 0.38$, $F = 8.0$, $p = 0.014$), and the IQR has not increased significantly (t-test on slope, $p = 0.19$; $R^2 = 0.16$, $F = 2.5$, $p = 0.14$).

Figure 2 shows the median rank and score for each answer document separated by year and with Dorothy Dixers separated out from the other questions. The score is the similarity score computed by the Zettair implementation of the Okapi BM25, and has no absolute meaning, but can be

**Table 1: Number of question-answer pairs for each year in the corpus (DD signifies Dorothy Dixer), and length of question and answer documents in words (after stopping).**

| Year | 1998 | 1999 | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| No. non-DD | 493 | 679 | 674 | 463 | 611 | 600 | 473 | 624 | 640 | 451 | 641 | 587 | 474 | 472 | 340 |
| No. DD | 494 | 674 | 668 | 461 | 595 | 567 | 476 | 626 | 634 | 443 | 641 | 576 | 466 | 403 | 281 |
| Total | 994 | 1363 | 1348 | 926 | 1209 | 1169 | 957 | 1265 | 1287 | 900 | 1284 | 1165 | 941 | 880 | 622 |
| Answers | | | | | | | | | | | | | | | |
| Mean Len. | 168 | 157 | 181 | 198 | 190 | 178 | 182 | 138 | 144 | 181 | 217 | 245 | 220 | 192 | 154 |
| Max Len. | 503 | 718 | 563 | 763 | 771 | 635 | 533 | 375 | 442 | 498 | 777 | 839 | 620 | 326 | 306 |
| Questions | | | | | | | | | | | | | | | |
| Mean Len. | 32 | 30 | 31 | 31 | 31 | 32 | 30 | 29 | 31 | 26 | 21 | 25 | 25 | 25 | 25 |
| Max Len. | 191 | 114 | 157 | 91 | 126 | 146 | 141 | 132 | 159 | 133 | 130 | 137 | 158 | 81 | 100 |

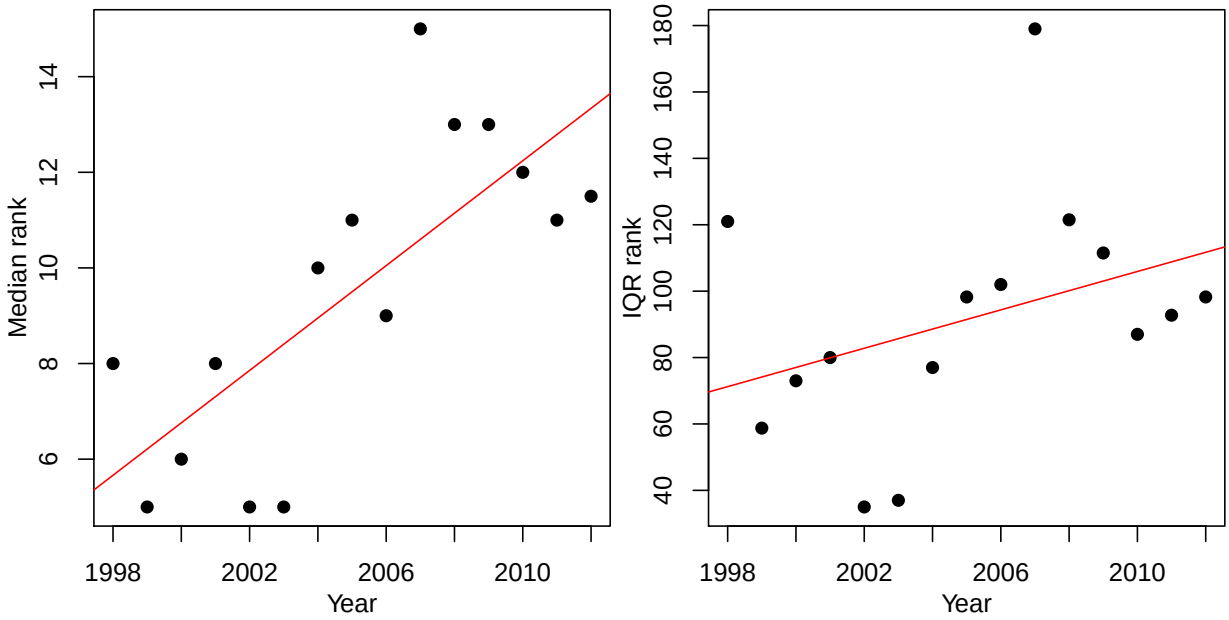



**Figure 1: Median and inter-quartile ranges (IQR) of the rank of the answer document for each question.**

compared relatively within a corpus. Again, lines shown are simple linear regression lines. Surprisingly, Dorothy Dixers show a significant increase in rank ($p = 0.002$) and decrease in score ($p < 0.001$) over time. Questions that were not Dorothy Dixers did not show a significant change in rank ($p = 0.454$), nor score ($p = 0.030$). This provides some evidence for Rodrigues' claim that Dorothy Dixers are increasingly used as a platform for making party political statements that have little relevance to the question.

## Questions in Writing

As an alternate experiment, the XSL scripts were modified to extract "QUESTIONS IN WRITING" from Hansard. The post processing scripts were also altered as the Questions in Writing had no interjections, and a slightly different introductory sentence. 7032 question-answer pairs were extracted from the corpus, covering years 2004 onwards as described in the first row of Table 2. As can be seen in the final three rows of the table, the IR engine ranks the answer to a question in position one generally over 50% of the time: far more success than on questions from Question Time, where the IR engine only gets about 20% of matching answers in rank position one. This seems to match public opinion that questions and answers in Question Time have less informative content than general parliamentary debate.

# 4. NEAR DUPLICATES

It is conceivable that some answers are similar to another, and so it may be unfair to expect an IR system to find exactly the right answer to a specific question from a pool of documents that are very similar. In fact, 59 answer documents are one word, "No.", and 8 are "Yes". Browsing documents of 15 words or fewer reveals that very few of them would contain terms that would match a specific question using any TFxIDF style IR metric: many are simply wordy ways of saying "no". For example, "The answer to the honourable gentleman's question is no" or "In precise answer to the tail end of the question: no". There are 251 documents with 15 words or less.

Taking this one step further, if an answer is run as a query against the collection of answers, where will itself rank? If the answer is unambiguous according to the IR engine, then

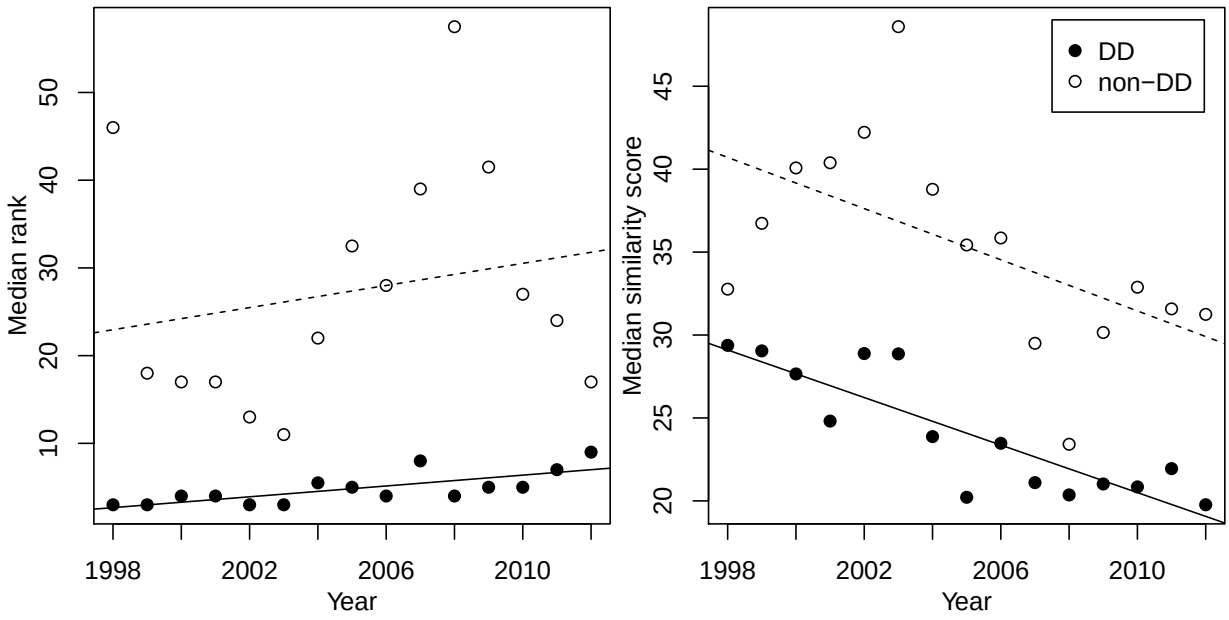



**Figure 2: Median and inter-quartile ranges (IQR) of the rank of the answer document for each question separated by Dorothy Dixer (filled) and non-Dorothy Dixer (open).**

**Table 2: Statistics on the Questions in Writing corpus. The last three rows show a summary of ranks of the answer documents as ranked by Okapi BM25 when the question is posed as a query.**

| Year | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 |
|---|---|---|---|---|---|---|---|---|---|
| Number | 5 | 1841 | 1560 | 1205 | 303 | 588 | 279 | 496 | 330 |
| Med. Rank | 2 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 |
| IQR Rank | 1 | 12 | 16 | 10 | 3 | 4 | 3 | 5 | 9 |
| % at Rank 1 | 40% | 50% | 47% | 44% | 58% | 55% | 64% | 58% | 53% |

it should appear at rank position one. There are 91 documents where this is not the case. These documents have an average length of about 8 words, with the longest being 22 words.

A similar exercise can be undertaken for questions. Building a document collection of just the questions as documents, and then running each question as a query, 44 do not come up with themselves as the first ranked document, 39 of which are Dorothy Dixers. The majority of these questions are of the form: "Would the minister please update the House about...". Perhaps some more sophisticated phrase stopping technique might distill these questions to their essence, but this was not tried in this work.

If we remove all question-answer pairs where one or the other does not return itself as the top ranked document, or the answer is 15 words or less, this excludes 386 pairs of questions and answers. Not surprisingly, this made no difference to the trend in median ranks or IQRs shown in Figure 1.

## 5. PERSONNEL

Figure 3 shows the median and IQR of ranks for each sitting period (typically about twenty days). One could conjecture that the 39th and 40th parliaments (prior to November 2004) contained Question Times that were more informative than that of the 41st, 42nd and current 43rd parliament post November 2004. One could also argue that the Prime Minister may not be responsible, as Howard was PM both before and after November 2004. Perhaps the quality of Question Time as an information exchange is driven by the Opposition Leader, who is the chief questioner, or the Speaker of the House, who chairs the debate and monitors content.

Figure 5 shows the ranks of returned answers grouped by the two people over the lifetime of the corpus. It is not that illuminating: no one obviously stands out as a culprit. Kim Beazley is the only person in the corpus to be opposition leader twice, and there is a significant increase in ranks (Wilcoxon $p < 0.001$) between his two terms, perhaps indicating that he alone did not control the content of Question Time in that period.

Perhaps the only observation that can be drawn from this part of the analysis is that the 6 sessions from February 2002 to December 2003 (Prime Minister Howard, Opposition Leader Crean, Speaker Andrew) had low ranks relative to other periods where personnel remained unchanged.

## 6. ON MESSAGE

It is currently common practice for parliamentarians from the major parties to be issued with "talking points" for a day, and there is a strong emphasis to "stay on message". Perhaps, therefore, the answers given to questions, particularly Dorothy Dixers, are very similar, making it difficult for an IR engine to pick the correct answer to any given question. To explore this possibility, we compared the scores of

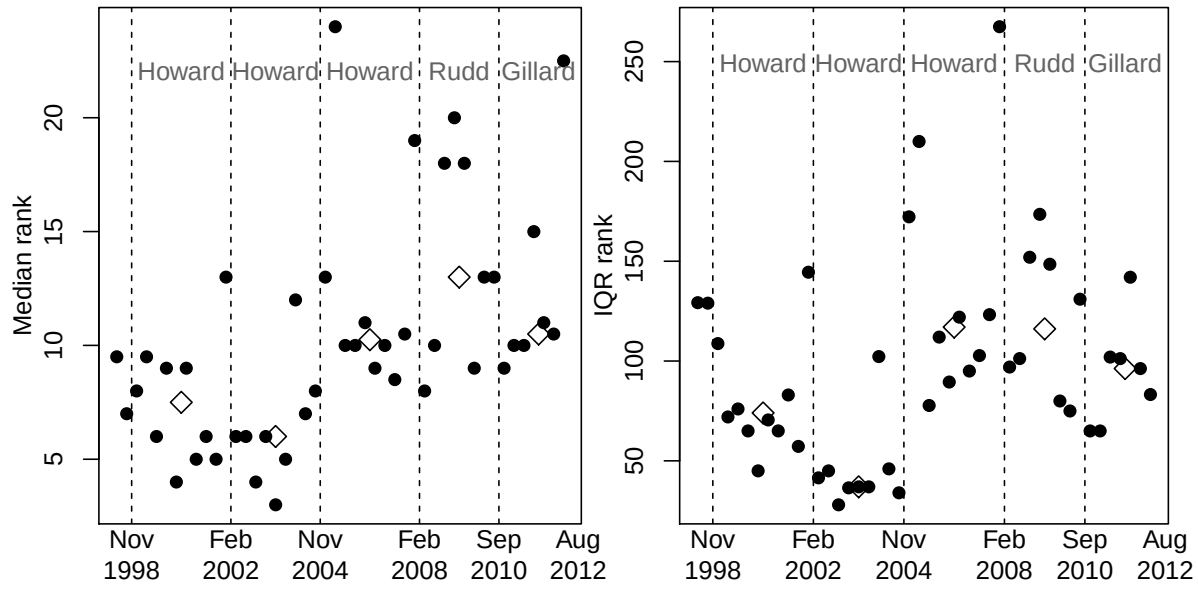



**Figure 3: Median and inter-quartile ranges (IQR) of the rank of the answer document for each question with one circle for each sitting period. Dotted lines show a change in parliament, with the names of the Prime Minister at the top of each parliament. The diamond gives the median value over the whole parliament.**

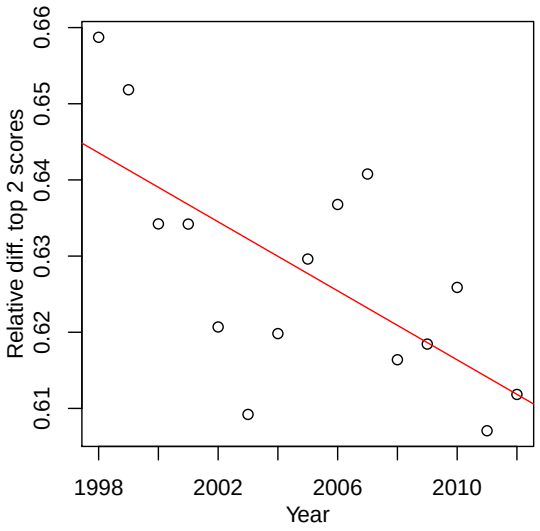



**Figure 4: Median relative difference in the Okapi scores of the top two answer documents returned when each answer is run as a query.**

the top two ranked documents when using each answer as a query.

For each answer document $a$, define $\Delta_a$ to be the difference between the top two scores of documents returned when $a$ is issued as query, divided by the top score. Thus if $\Delta_a$ is high, the second document is scored as very different from the first, while a low $\Delta_a$ indicates that the top two documents are similar, according to the IR engine's similarity measure. There is no correlation, between the median rank of answers as returned by the IR engine in a parliamentary period and median $\Delta_a$ (Pearson, $p = 0.4$). But, there is a linear trend for median $\Delta_a$ decreasing from about 66% in 1998 to 61% in 2012 ($p = 0.007$). Figure 4 shows the data.

Splitting the data into Dorothy Dixers and non Dorothy Dixers reveals little. Both follow a similar pattern as the aggregated data, decreasing from bout 66% to 61%.

## 7. DISCUSSION

This paper attempts to objectively measure the quality of Question Time in the House of Representatives of the Australian Federal Parliament using state-of-the-art document ranking schemes that are the underlying technology of modern search engines. All answers given in Question Time in the period 1998 through to August 2012 are collected and indexed by the IR system, then each question is posed as a query to the system and the rank of its known answer recorded. The IR engine has more trouble finding answers to questions in recent years than in the past, leading to the conclusion that Question Time has reduced in quality over the last decade.

Throughout the experiments the Okapi BM25 metric [7, 8] with default settings was used as the ranking metric. We also generated Figure 1 using a Language Model scheme [6] with Dirichlet smoothing [13] using $\mu = 2000$ and $\mu = 500$, but there was no appreciable difference from the Okapi results, hence they are not reported in this paper. Perhaps there are other similarity metrics that are more suited to this retrieval task. Going beyond the standard approaches to include the meta-data identifying speakers, parties, and so on, or even exploiting the audio recordings of question time to determine speech patterns would be interesting areas for future study.

While the length of questions in words has decreased in recent years (see Table 1), the marked increase in median rank of answers occurs for 2003 where question length was still on a par with previous years. It isn't until 2007 that

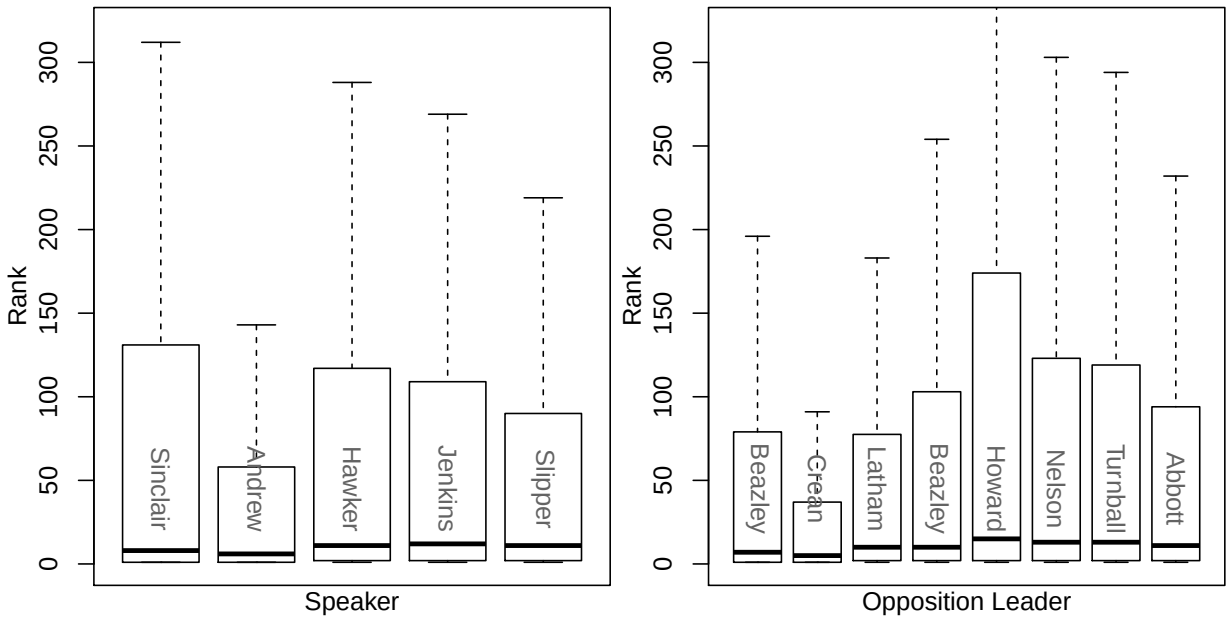



**Figure 5: Ranks of answer documents for each question grouped by Speaker (left) and Opposition Leader (right).**

questions become noticeably shorter. It is unlikely, therefore, that the number of words in questions or answers can account for the results.

Obviously using an IR system to measure quality of Question Time has its limitations. All of the IR approaches tested use stemming, stopping, treat questions and answers as bags-of-words, and ranks documents using a TFxIDF scheme. Such a scheme increases the similarity score between a document and a query if query words that occur frequently in a document (TF) and do not occur frequently in other documents (IDF). This approach is currently the best available technology for efficiently matching queries to documents [2]. It does not, however, incorporate the myriad of nuances that are present in Question Time question and answers. Politicians include doublespeak, metaphor, half-truths, red herrings, retorts to interjections, and any other manner of rhetorical and Vaudeville devices in their questions and answers. Also the "documents" used here are transcripts of the spoken word, so include social pleasantries (or ugliness) that it not usually found in written documents. The TFxIDF ranking used here captures none of this variation. As an aside, note that research into Spoken Word Retrieval [5] has focused predominately on the speech recognition side of the process, and not on the IR component. This seems like a fertile area for future research using this corpus.

In order to validate the use of TFxIDF rankings as a measure of quality of Question Time data, human judgements of question-answer pairs are required. Given all of the complications of the information exchange in Question Time, annotating this data set requires careful planning and instruction to annotators. We will explore this avenue in future work.

Any judgement of answer quality depends on what one sees as the aim of Question Time. If you subscribe to Mary Crawford's observation [3] that "For better or worse, it represents the translation of the political process into the everyday", then reducing the content of Question Time purely to an information matching problem is a futile exercise. If one would rather that an hour of an Australian politician's life was used in information sharing, rather than theatre, then the results in this paper directly apply. There is most likely a middle ground, where theatre does not obstruct information sharing, the text of which presents a major challenge to current automatic language analysis and information retrieval systems, and also a significant challenge to any manual annotation of the data.

Regardless of the theatrical content of Question Time, reduction of the repetition of information may be welcome. The results in Section 6 show that the information content of the answers to Dorothy Dixers are becoming more similar to each other as time goes on, and this could be the primary cause of the increased median rank of answer documents. Further investigation is required.

## 8. CONCLUSION

The quality of information exchanged in Question Time, as measured by an IR system, has decreased from 1998 to 2012. There has been a sharp deterioration since the end of 2003 that continues through to 2012. Surprisingly, the information contained in Dorothy Dixers has decreased over this period, while the information in other question-answer pairs has remained consistently poor when compared with questions in writing.

## 9. ACKNOWLEDGMENTS

Thanks to Tim Baldwin, Alistair Moffat and Falk Scholer for helpful discussions. Thank you to the anonymous reviewers for useful comments. This research is supported by ARC Grant FT0991326.

# APPENDIX

## A. TECHNICAL DETAILS OF CORPUS CONSTRUCTION

Figure 6 shows the final XSL used. Prior to March 2011, interjections were tagged separately as `<interjection>` tags, simplifying the XSL considerably. Figure 7 shows the relevant XSL.

Post processing of the extracted data proceeded as follows.

1. TREC-style [11] tags were inserted to demarcate questions and answers.

2. The "name (electorate) (time):" prefix of all questions and answers were removed.

3. The first sentence was removed from all questions that contained more than one sentence. Typically the first sentence is of the form "My question is to ...". While informative, this project is interested in the content of the question, and not the intended recipient.

4. The phrase "Members interjecting" and its many variants was removed from all questions and answers using the `sed` script shown in Figure 8.

5. During a question, if a member is interrupted, their name appears in the question after their continuation in upper case and followed by a colon. This was removed from all questions with the following `sed` script.

```
s/Mr [A-Z]*://g
s/Mrs [A-Z]*://g
s/Dr [A-Z]*://g
```

6. The phrase "(Time expired)", with possible punctuation between the 'd' and '(', was removed from all questions and answers.

7. Extended ASCII characters were mapped to standard ASCII characters in both questions and answers. In particular: the Euro and Pound symbols were mapped to `EURO` and `GBP` respectively; various length hyphens were all mapped to a simple minus sign; and various quotation marks were mapped to simple single quotes. Double quotes `"` were also substituted with a single `'` to prevent phrase processing by the IR engine.

## B. REFERENCES


[1] AAP. Squabble breaks out over parliamentary reform. *Sydney Morning Herald*, Sept. 2010.

[2] T. G. Armstrong, A. Moffat, W. Webber, and J. Zobel. Improvements that don't add up: ad-hoc retrieval results since 1998. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 601–610, New York, NY, USA, 2009. ACM.

[3] M. Crawford. Question Time: don't change the contest we want to watch. `http://theconversation.edu.au`, Sept. 2012.

[4] `http://www.aph.gov.au/Parliamentary_Business/Hansard`.

[5] M. Larson and G. J. F. Jones. Spoken content retrieval: A survey of techniques and technologies. *Foundations and Trends in Information Retrieval*, 5(4-5):235–422, 2011.

[6] J. M. Ponte and W. B. Croft. A language modelling approach to information retrieval. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proc. ACM SIGIR*, pages 275–281, Melbourne, Australia, 1998.

[7] S. E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society For Information Science*, 27:129–146, 1976.

[8] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In W. B. Croft and C. J. van Rijsbergen, editors, *Proc. ACM SIGIR*, pages 232–241, Dublin, Ireland, 1994.

[9] M. Rodrigues. Parliamentary reform. `http://www.aph.gov.au/About_Parliament/Parliamentary_Departments/Parliamentary_Library/pubs/BriefingBook43p/parliamentaryreform`, 2010.

[10] M. Turnball. Republican virtues: truth, leadership and responsibility. *The George Winterton Lecture 2012 delivered at The University of Western Australia*, Sept. 2012.

[11] E. M. Voorhees and D. K. Harman. *TREC: experiment and evaluation in information retrieval*. MIT Press, 2005.

[12] http://www.seg.rmit.edu.au/zettair, 2012.

[13] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 334–342, New York, NY, USA, 2001. ACM.

```
<xsl:for-each select="//debate[debateinfo/title='QUESTIONS WITHOUT NOTICE']/subdebate.1[question and answer]">
    <xsl:for-each select="question">
        <xsl:if test="count(preceding-sibling::question) = 0">
            <xsl:for-each select="node()//span[@class='HPS-MemberQuestion']">
                <xsl:for-each select="ancestor::body/p">
                    <xsl:if test="not(.//span[contains(@class, 'Interjecting') or contains(@class, 'Office')])">
                        <xsl:value-of select="."/>
                    </xsl:if>
                </xsl:for-each>
            </xsl:for-each>
        </xsl:if>
    </xsl:for-each>
</xsl:for-each>
```

**Figure 6: The XSL used for extracting the first question in each subdebate from Hansard XML published after March 2011. Answers are extracted similarly, replacing "question" with "answer" in line 3.**

```
<xsl:for-each select="//debate[debateinfo/title='QUESTIONS WITHOUT NOTICE']/subdebate.1[question and answer]">
    <xsl:for-each select="question">
        <xsl:if test="count(preceding-sibling::question) = 0">
            <xsl:for-each select=".//para">
                <xsl:if test="not(./ancestor::interjection)">
                    <xsl:value-of select="."/>
                </xsl:if>
            </xsl:for-each>
        </xsl:if>
    </xsl:for-each>
</xsl:for-each>
```

**Figure 7: The XSL used for extracting the first question in each subdebate from Hansard XML published before March 2011. Answers are extracted similarly, replacing "question" with "answer" in lines 2 and 3.**

```
s/Opposition members interjecting//g
s/Government members interjecting//g
s/Honourable members interjecting//g
s/Opposition member interjecting//g
s/Government member interjecting//g
s/Honourable member interjecting//g
s/A government member interjecting//g
s/An opposition member interjecting//g
s/Mr [a-zA-Z'"']* interjecting//g
s/Mrs [a-zA-Z'"']* interjecting//g
s/Ms [a-zA-Z'"']* interjecting//g
s/Dr [a-zA-Z'"']* interjecting//g
s/Mr [a-zA-Z]* [a-zA-Z'"']* interjecting//g
s/Mrs [a-zA-Z]* [a-zA-Z'"']* interjecting//g
s/Ms [a-zA-Z]* [a-zA-Z'"']* interjecting//g
s/Dr [a-zA-Z]* [a-zA-Z'"']* interjecting//g
s/The member for [A-Za-z]* is interjecting//g
s/The member for [A-Za-z]* [A-Za-z]* is interjecting//g
s/The member for [A-Za-z]*-[A-Za-z]*is interjecting//g
```

**Figure 8: Sed script used to remove interjections that were not tagged in the XML.**