Re-ranking the Search Results for Users with Time-periodic Intents

Gina Horscroft
Department of Computer Science
University of Cape Town
Cape Town, South Africa
HRSGIN001@myuct.ac.za

Jivashi Nagar
Department of Computer Science
University of Cape Town
Cape Town, South Africa
jnagar@cs.uct.ac.za

Hussein Suleman
Department of Computer Science,
University of Cape Town
Cape Town, South Africa
hussein@cs.uct.ac.za

ABSTRACT

This paper investigates the time of search as a feature to improve the personalization of information retrieval systems. In general, users issue small and ambiguous queries, which can refer to different topics of interest. Although personalized information retrieval systems take care of user's topics of interest, but they do not consider if the topics are time periodic. The same ranked list cannot satisfy user search intents every time. This paper proposes a solution to rerank the search results for time sensitive ambiguous queries. An algorithm "HighTime" is presented here to disambiguate the time sensitive ambiguous queries and re-rank the default Google results by using a time sensitive user profile. The algorithm is evaluated by using two comparative measures, MAP and NDCG.

Results from user experiments showed that re-ranking of search results based on HighTime is effective in presenting relevant results to the users.

CCS CONCEPTS

• **Information systems** → **Query intent**; *Information retrieval diversity*;

KEYWORDS

Time-periodic queries, Re-ranking search results

ACM Reference Format:
Gina Horscroft, Jivashi Nagar, and Hussein Suleman. 2018. Re-ranking the Search Results for Users with Time-periodic Intents. In *Proceedings of SAICSIT conference (SAICSIT'18).* ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The Web is the most used source of information these days. With its exponential growth, everyday new pages are being added to it. It contains information on almost every topic that a user may look for. Moreover, surfing the Web seems easy to everyone, including those who are not computer literate. Search engines have made their job easy. But the ambiguity inherent in common languages like English is a problem. There are many words that may point to the same concept and there is one word that refers to many concepts. For example, the words "beauty" and "pretty" refer to the same quality of attraction of anything, while the word "java" may point to "programming language" or "coffee" or may be the "island".

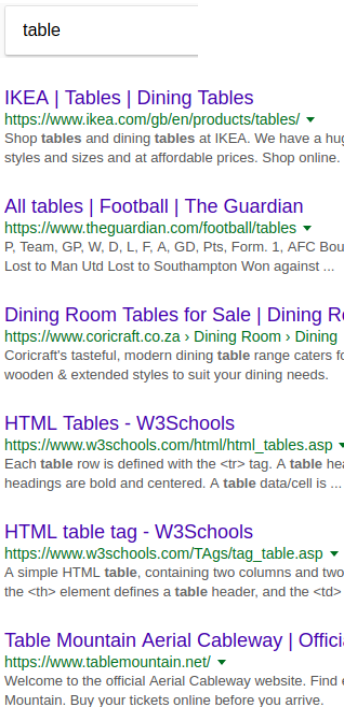
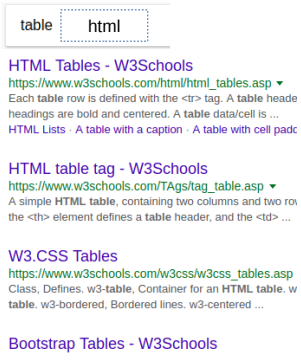


Figure 1: Search results for an ambiguous query "table"

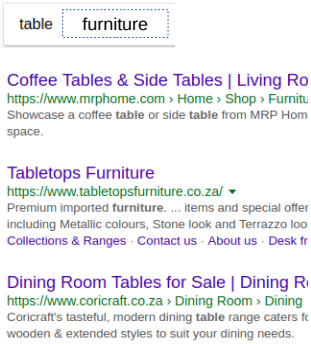
Search engines have to deal with all kinds and levels of ambiguities in order to find relevant results. If a search engine provides search results based on one of the latent topic that is more popular than the other topics associated with the ambiguous query term, there is a chance that these results satisfy some users but not all of the users can find them relevant. The results found on the first page or the second page may not be the results that user is looking for [7]. This approach may result in unsatisfied search experience. As shown in Figure 1, if a user makes a search for query "table", the default results are related to different topics maybe because

SAICSIT'18, September 26-28, 2018, Port Elizabeth, South Africa

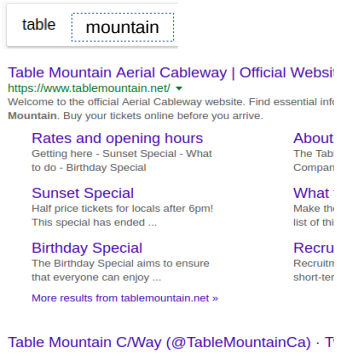
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn



(a) Search results relevant to latent intent "HTML"



(b) Search results relevant to latent intent "furniture"



(c) Search results relevant to latent intent "table mountain"

Figure 2: Latent intents of ambiguous query "table"

the query "table" is ambiguous in nature. But if the search engine knows the latent intent of the user query, better results could be provided as shown in Figure 2. For example, if user wants to learn about "HTML tables", the more relevant result set would be the one shown in Figure 2a. Figure 2b shows the results that could be more relevant if the user is looking for "furniture". Very few results could be found on the first page related to query intent about one of the visiting places in Cape Town, the Table Mountain. The more relevant and informative results for a user who is looking forward to visit the Table Mountain would be the one shown in Figure 2c. This issue of understanding user intent is known as the multiple intents re-ranking problem. It was first introduced by Azar et al. [2], where it was noted that an accurate ranking model should assume that queries can have multiple intents, to minimize the effort of a user to find relevant results. Search engines can improve the user's search experience if personalization is implemented [31]. To know user's personal search interests, two approaches can be employed, ie. explicit feedback and implicit feedback. According to Fox et al [3], getting explicit feedback from the users is hard. Users may not be interested to give the explicit feedback as it causes extra burden on them. Moreover it is not cost effective [8] as compare to the implicit feedback. User's intents can be learned implicitly from click history or log analysis.

It has been seen that a user's topics of interest vary over time [20], [18]. For example, a Computer Science student searches for the topics related to Computer Science during his study time but, during leisure time, he may search for some coffee shops. The top ranked results are mostly found to be biased towards the popular intent associated with the ambiguous term. So, if he/she searches for "java" during his/her study time and leisure time, the result would be the same. That ranking order may satisfy his/her search intent at one time but definitely not the other time. To satisfy the other time intent, he/she may have to go down the list, maybe the next pages or he/she has to frame a new query.

This study investigates the use of "time of search" as a feature in a ranking algorithm and proposes a solution to address a user's time-periodic ambiguous (TpA) queries. The user profiles with time-periodic queries were assumed to evaluate the proposed re-ranking algorithm.

2 RESEARCH QUESTION

This study aims to answer the following research question:

Can a Web-search ranking algorithm that personalizes results on the basis of time-periodic user profiles return results that are more relevant to a user than an algorithm that does not?

After consulting the relevant literature, and to the best of our knowledge, it appears that no personalisation algorithms factor implicit query time data into re-ranking. To address this issue in this paper, results returned by popular search engines (specifically Google) are re-ranked based on a time-periodic user profile. Since the time periodic user profile is a model of the topics that interest a user and times when these topics are relevant, our re-ranking algorithm factors in implicit query time data into the re-ranking process.

The rest of the paper will give details about the study. Section 3 will give an overview of related work. Section 4 will describe

the proposed re-ranking algorithm "HighTime" and will cover Experimental Setup, followed by evaluation methodology in section 5. Section 6 analyse the results and their statistical significance. Section 7 is discussion and Section 8 is conclusion and future work.

3 RELATED WORK

3.1 Ambiguity in IR

Studies have already been done focusing on the ambiguity problem in information retrieval. Sanderson [22] investigated the scarcity of ambiguous queries in test collections. Traditionally, test collections contain only one interpretation per topic. He described a method for creating test collections containing ambiguous queries from existing resources like disambiguation pages in Wikipedia, that link all interpretations of the article title together. For example, the disambiguation page for "Java" contains multiple meanings of the word, from the programming language, to locations, to music, to coffee. Sanderson created a data set using words/phrases from the Wikipedia disambiguation page, and merged this with an existing test collection. It was found that these queries had a significant negative impact on the performance of conventional IR systems. Some papers have discussed the type [28],[9] of ambiguous queries. Song et al [28] analysed and studied a 12 days sample log from Live Search and found three different categories of queries ranging from ambiguity to specificity ie. a query that has more than one meaning (ambiguous query); a query that covers a number of subtopics (broad query); and a query with clear meaning with specific topic. Another recent study [9] has presented a method to classify the ambiguous queries using post search by applying content similarity approach. They considered contextual and temporal features from the Web results of different ambiguous queries. A number of studies have contributed in query subtopic mining [36], [32]. Yi et al [36] proposed the use of a tripartite graph based on user search behaviour on search log data and non negative sparse LSA model to mine the query subtopics. On the other hand, Ullah et al [32] used word embedding and a short-text similarity measure to mine the query subtopics. To satisfy a number of users with different search intents, diversification of search results has been considered by many studies. Shajalal et al. [24] proposed a method to diversify subtopics related to a user query. This study utilized query suggestion for a user query from various search engines as the underlying subtopic candidates of that query. These subtopics are then clustered considering their semantic, lexical and popularity based features. Kumar et al [12] considered a Web page similarity feature to diversify the results. According to this study, clustering using Web page similarity feature retrieve relevant as well as diverse results. Lesk [14] took a simple approach to disambiguating words, on the assumption that words in the area of text of the ambiguous word (the "neighbourhood") will share a topic. Krovetz and Croft [11] examined the ambiguity of words in test collections, and found that disambiguation of queries could be performed quite effectively using dictionary lookups and thesauri. A similar idea is used in the proposed algorithm, HighTime, to attempt to disambiguate queries by expanding the keywords used in the altered [14] algorithm by their synonyms. But the difference is in corpora. Rather than using existing corpora for testing, the real time top ranked results returned by Google for a set of ambiguous queries were used.

All these studies have aimed to disambiguate the query to provide best relevant or matching content to the user but none of them focused on time-periodic ambiguous queries.

3.2 Re-ranking and Time-sensitivity

If a search engine provides search results based on one of the latent topic that is more popular than the other topics associated with the ambiguous query term, there is a chance that these results satisfy some users but not all of the users can find them relevant. As seen in the literature, some studies believe that it is better to cover all the latent topics related to that query in the result set [23],[29]. This diversity sometime becomes an information overload and leads the users to struggle in order to find relevant information from the Web. As shown in Figure 3, if a user searches for query "cup", the results retrieved would be related to different topics like "World Cup", "Crockery" etc. as shown in the grey box of the Figure 3. If these results are rearnked in a particular order in effort to cover the diversity of the topics as shown in the left-hand side box of the Figure 3, it does not make much difference in making the user's search experience better.

In the literature reviewed, two main approaches were taken in personalizing the search results. The first approach modifies the actual query, while the second approach re-ranks the returned results using information in user profile. The query modification approach is less effective than the re-ranking approach as the information about the user found to be relevant is lost after the search query session ends [16]. As a consequence, the query modification approach does not adapt to a user's interests as the user provides more information about what he/she finds interesting.

The traditional ranking algorithm based on Web content is not sufficient to retrieve relevant and useful information to the users [7]. PageRank is the base of the ranking algorithm employed by the Google. Most of the studies in this field have tried to improve on this. One such study [6] presented Topic sensitive PageRank. This algorithm gives a different ranking score to each page based on its topic. The degree of importance of the topic to the search query is considered to compute the relevance of a Web page. According to Agichtein et al [1], incorporating implicit user feedback improve the Web search performance with popular content and link based algorithms. The implicit features considered are dwell time, scroll time and query reformulation patterns along with click-through behaviour. They found implicit feedback important for queries with poor original ranking of results.

To address an issue often forgotten in this field, Vu et al [35] modelled search tasks that take time into account. Efficiency of this approach was then tested by comparing the order of relevance delivered by a search engine before and after being re-ranked, factoring in time. This personalisation technique takes the results generated using the commercial search engine and then re-ranks them using the time-sensitivity algorithm-TimeTask. This study sees time-aware in a sense that more recent documents may be more relevant to a user. As it can be seen in right-hand side box of Figure 3, the results are reranked according to the recency. Results showed that the performance of the TimeTask algorithm was better than the default search-engine results. But if a user is looking for crockery, majority of the top ranked results will not prove relevant

to him/her. If a user is not interested in recent events but rather some old time events, he/she may not be helped by recency based re-ranking order. Thus, re-ranking based on recency may not prove beneficial in all circumstances. Kanhabua and Nørvåg [10] proposed a number of methods to determine the time of implicit temporal queries. Implicit temporal queries are those that have a latent time feature, for example, "Germany FIFA world cup". This query clearly means the world cup event in 2006. The time of the query can be used to improve the relevance of results.

Lee and Kim [13] applied and tested the effectiveness of a click model for time-sensitive queries. According to them, several click-modelling techniques exist, but these cannot be applied directly to temporal queries where it may instead degrade the relevance of queries. Using general click models as feedback for search quality may not be accurate for temporal queries because even if a search was relevant at one point in time it may not be relevant at another. Volume trends for a week showed spikes at certain points in time where relevance may be higher. This showed that certain queries have more weight at different times. The model was made up of two parts: a click model to calculate a relevance score for documents, and the turning point to determine which data to use. The turning point is determined using a sliding-window method to determine the last date where search volume was 1.5 times more than the average for the last 5 days. Results showed that this model performed best in terms of the average normalized discounted cumulative gain.

4 HIGHTIME: DESIGNING AND EXPERIMENTAL SET-UP

According to Seig et al [26], there are two challenges in retrieving relevant information from the Web. One is to identify specific user context and other one is to organize the information according to that context. This paper proposes to disambiguate the search queries using time of search along with query word expansion. For the experiment, a set of four ideal "simulated" user profiles were created to evaluate and test of the re-ranking algorithm. These user profiles were created by modelling different interpretations of the same/similar queries at different times of the day. The user profiles were analysed for existing topics. To rank the search results of time-periodic ambiguous queries (TpA) for a user, ranking algorithm, HighTime, is developed. It considers the following factors:

(1) User Profiles: As indicated in our previous work [18], users search for different topics at different times, this study intends to use time-periodic user profiles using their implicit information ie. previous search queries and the snippets of the respective clicked results along with the time of search.
(2) Result Retrieval: The first 300 results for a query were collected by Web-scraping the results of a Google search. These results were then cleaned to remove non-ASCII characters and trailing ellipses. Results are returned in the ranked order provided by Google. Each result consists of a title, URL, and a snippet.
(3) Snippet Analysis: To determine the topic/topics of the results, two primary approaches are taken in traditional algorithms: document analysis, where the entire text of the document is analysed; or snippet analysis. Document analysis is time consuming as compared to snippet analysis. Due to the high number of results that were required to be analysed in the re-ranking algorithm, the choice was made to analyse snippets, not entire documents. Snippets contain the main summary of the documents. To extract the keywords, Rapid Automatic Keyword Extraction (RAKE) algorithm [21] is implemented. RAKE is an unsupervised approach to key phrase extraction. The intent of this was to summarize what the primary topics, phrases or keywords of a result are, so that this could be matched with the modelled topics in the user profile.
(4) Dictionary Expansion: The keywords/phrases for each result are then passed on to the synonymic expansion to include the word(s) having the same or closely the same meaning using WordNet [4]. WordNet is an English lexical database that can be used for natural language processing, and provides extensive coverage for ambiguous words [22]. Sets of synonyms – synsets - are calculated for each keyword. These sets consist of synonyms and approximations of meaning for words and phrases to that keyword.
(5) Score Calculation: The topic match for each entry in the profile was determined using word-count overlap between topics in the user profile, and the synset of key phrases sourced from the snippet for each result. Each topic in the user profile had one or more associated TimeFactor, TF . The TimeFactor is a normalized representation of time. This was modelled at a scale of 24 hours. Time normalisation was done to easily compare different times of the day. For example, if a search is made at 10:00, the topic relevant at 9:00 according to a user profile is more relevant at 10:00 as compared to the topic that is relevant at 23:00. The final score is assigned to each result after matching with the user profile and is made up of a combination of Overlap (topic match), time-weighting, and positional normalization (explained in the next section).
(6) Result Re-ranking:
The results are re-ranked in descending order of score, stored and compared with the original rankings.

To know the topics of interests of a user, in a similar way to Lesk's work [14], HighTime calculates the overlap between words in a result's snippet, and those contained in topics in the user profile. The following example illustrates how this would translate to form part of HighTime algorithm.

The user illustrated by Figure 4 often searches for topics pertaining to coffee in the morning, before the workday, while during the workday later in the afternoon he/she often search for topics pertaining to programming. The query "Java" would return many results that may belong to different topics. The snippets for two such results may be:

(1) Java can reduce costs, drive innovation, and improve application services as the programming language of choice for IoT, enterprise architecture, and cloud ...
(2) Coffee has many names. Some, such as "espresso," and "drip" refer to how coffee is made. Others, such as "mocha" and "cappuccino," refer to a specific beverage made with coffee. Still others refer to coffee's origins and history.

An entry in the user profile contains the key words "Coffee", "Beverage" and "Drink". This entry would have more overlap with

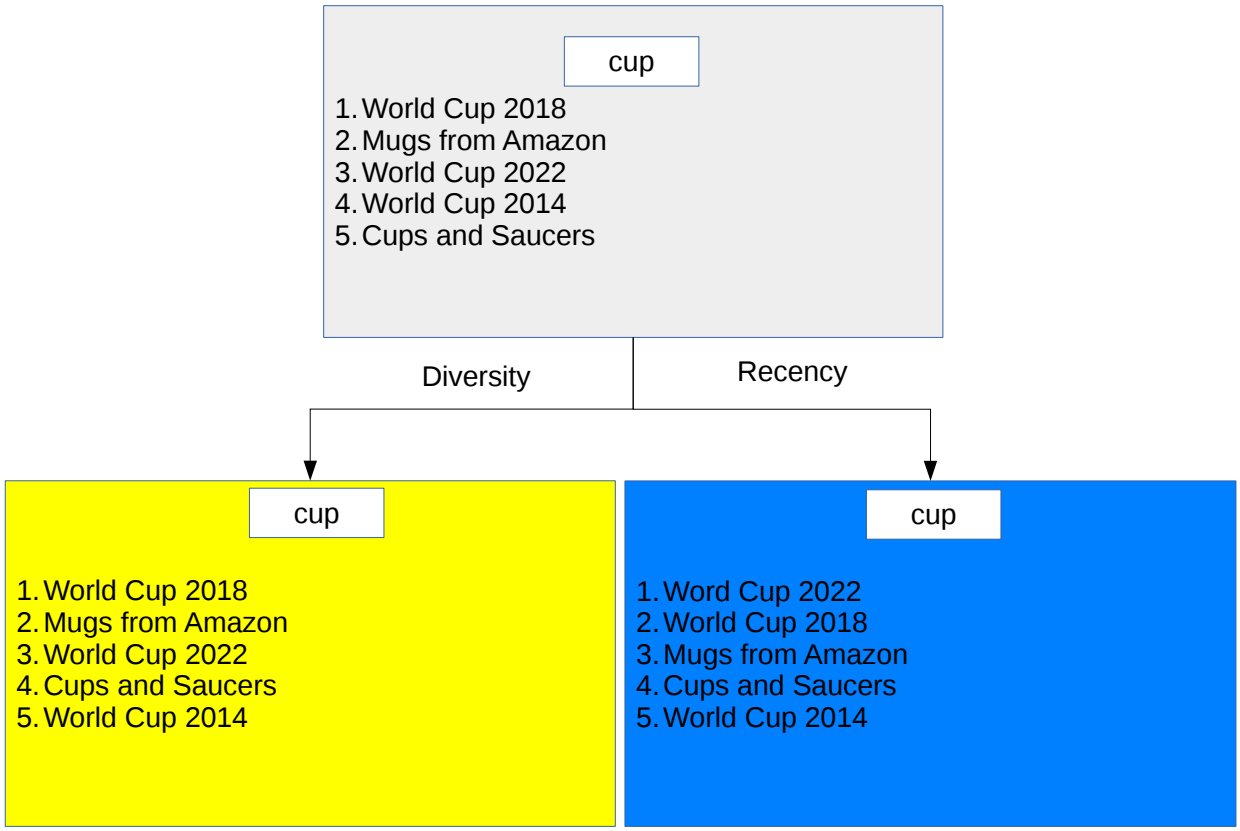


Figure 3: Default results re-ranked according to Diversity and Recency

```
# FORMAT: [<TimeFactor>, ..., <TimeFactor>] [(<Probability>, <Topic>), ..., (<Probability>, <Topic>)]
[0.27] [(0.6, Coffee), (0.6, drink)]
[0.59, 0.68] [(0.2, Programming), (0.6, Java), (0.6, Reference)]
```

Figure 4: Example of a sample user profile

the second result snippet. The HighTime algorithm would therefore assign a higher *Overlap* score to the second result. Along with Overlap, it also considers the time of search. The topics frequently searched at current time get higher *Weighting* score.

Equation 1 shows the calculation of the TimeFactor.

$$TF = \left| \frac{Hours * 60 + Minutes}{1440} \right| \quad (1)$$

where TF is the TimeFactor.

For each TimeFactor in each topic in the user profile, a Weighting is calculated to determine which topics are more relevant at the current time of query. The formula for this Weighting is shown in Equation 2. The higher a Weighting, the closer of a match to the current time - and therefore a higher interest in topics at that entry in the user profile. This is an indication of the relevance of topics at certain times.

$$W = 1 - min(|TopicTF - CurrentTF|, 1 - |TopicTF - CurrentTF|) \quad (2)$$

where W is the Weighting.

A threshold value was determined through trial and error to only give weight to topics above a certain value. For entries in the user profile with a weighting of above a threshold value, t, where t = 0.6, a score for each result is calculated. This word-overlap score is multiplied by the weighting of the entry in the user profile (to give more weight to topics with higher weightings). Finally, a logarithmic normalization of the original ranking returned by the Google search factored the original ranks into the overall equation. This meant that, for two results with a similar calculated topic and time score, the result that was ranked higher by Google would be ranked higher in the HighTime ordering. This served to decrease the likelihood of having multiple results with tied scores [15]. Equation 3 shows the equation to calculate the score assigned to each result, where overlap and weighting are the values in the user profile entry.

$$FinalScore = \sum_{Entry=1}^{E>t} Overlap.Weighting.NormalizedRank \quad (3)$$

This value is calculated for every entry in the user profile with a weighting above the threshold value, and the summation of these scores is the final assigned score to that result.

Algorithm 1 Working of HighTime

```
for all results retrieved do
    Apply RAKE on Title and Snippet
    Dictionary expansion of keywords/phrases
    for all User profile entries do
        Calculate Overlap between topics in User profile and those
        in the synset of key phrases of result snippets
        Calculate Weighting using Equation 2
        if Weighting > threshold (t) then
            Calculate FinalScore with NormalizedRank using Equa-
            tion 3
        end if
    end for
end for
Re-rank results based on FinalScore
```

Algorithm 1 illustrated the steps of working of the proposed algorithm, HighTime.

The calculation of FinalScore of results for a user who has three different topics of interest is shown in Table 1. Each User Profile Entry represents a certain topic of interest of a user. Overlap is the number of keyword matches that exist for a specific topic entry in the user profile. The greater Overlap value shows more topic relevance. Weighting represents the TimeFactor weighting that shows how much relevant the topic is at the time of search. Normalised Rank value is the normalization of rank ie. rank of the result/number of ranked results. Here the result of rank 5 is considered in the table 1 and the total number of results considered is 10.

Table 1: Final Score Calculation for a result at the original rank of 5

Profile	Overlap	Weighting	Normalised Rank	Total
Entry 1	14	0.8		5.6
Entry 2	7	0.7	0.5	2.45
Entry 3	3	0.4		N.A.
Final Score				8.05

Note that the Profile Entry 3 in the user profile has a weighting less than that of the threshold value, and so, is not included in the final calculation.

5 EVALUATION METHODOLOGY

The algorithm was evaluated in an offline manner. This method of evaluation has the benefits of being fast, repeatable and cost-effective [17]. In offline user evaluation, users make judgments from a list of ranked documents produced prior to evaluation. 4 short queries were chosen for evaluation, with each participant evaluating the rankings for one of the queries. The queries were intentionally kept short based on existing literature [5][25][30] that indicates that short queries allow for ambiguity. One aim of HighTime algorithm was to remove this ambiguity using implicit information in the user profile and the explicitly stated time of query. Each list of ranked documents contained only the first 10 results for a query. This number was chosen based on research on user behaviour [33] that shows that 91 percent of users do not search further than the first page. Users prefer to attempt to rephrase the original query and try the search again. A good re-ranking algorithm should therefore achieve high relevance scores in those first 10 results. In total, 24 participants were recruited to make relevance judgments on the top 10 HighTime results and original Google results. 24 is a common choice of participants for information retrieval user evaluation [34][19]. Participants were given four sets of ranked documents in total: two showing the original order of results returned by the search engine, and the other two showing the order of results re-ranked using the HighTime algorithm. Participants were not informed of which ranking was which, to prevent bias. Participants made judgments for the original rankings and the HighTime rankings for two different interpretations of the ambiguous query. Each query was therefore evaluated by 6 participants. For example, for the query, "Java", 6 participants made judgments for both the original search results and the HighTime results, where the user intentions were: 1. Java: the programming language 2. Java: the coffee

The original search results contained the same order of documents for both interpretations of the query, while the HighTime results were re-ranked for two different time periods where the user would be interested in different interpretations. Participants were

informed of the query and the user intention, and asked to grade each result on the scale from 0-3. Table 2 shows the four ambiguous short queries for which users scored results returned by Google, and results re-ranked by the HighTime algorithm. These queries were sourced from Wikipedia disambiguation pages [27], with the user intention sourced from two such interpretations, provided by Wikipedia. Each query has a user intention at that time - modelled as part of the User Profile - that shows what results the user would find relevant at that time.

Table 2: Ambiguous short queries with their user intent

Query No.	Query	User Intent
1	java	The coffee
2	java	The programming language
3	apple	The technology company
4	apple	The fruit
5	jaguar	The animal
6	jaguar	The American football team
7	venus	The planet
8	venus	The Roman goddess

6 RESULTS AND STATISTICAL ANALYSIS

6.1 User Evaluation Results

Tables 3, 4, and 5 show the average MAP, DCG, and NDCG values for each of the 8 queries. The value for each query is the average metric (MAP, DCG, or NDCG) calculated from the scores of the 6 participants evaluating that query. Table 3 shows the MAP scores

Table 3: MAP scores for the HighTime and the original Google rankings

Query	HighTime	Original
1	0.75	0
2	1	1
3	1	1
4	0.99	0.03
5	0.95	0.04
6	0.97	0.99
7	1	1
8	0.79	0.22

for the 8 queries and interpretations for the HighTime and Google rankings. Table 4 shows the DCG scores before normalization by the ideal DCG for each query. DCG scores measure the usefulness of a ranking. As the entire Web-corpus of results cannot be manually evaluated to determine a perfect IDCG, DCG scores were tracked to ensure that the NDCG scores followed a similar pattern and that the IDCG was a good approximation of an ideal ranking (as the IDCG was calculated from manually assigned scores). Table 5 shows the NDCG scores for the 8 queries and interpretations for the HighTime and Google rankings. It follows similar pattern to

Table 4: DCG scores for the HighTime and the Original Google rankings

Query	HighTime	Original
1	10.44	0.0
2	15.04	14.72
3	14.07	14.7
4	10.37	0.05
5	11.11	0.17
6	14.39	14.20
7	15.14	15.42
8	8.49	0.65

Table 5: NDCG scores for the HighTime and the Original Google rankings

Query	HighTime	Original
1	0.70	0.0
2	1.0	0.99
3	0.95	0.98
4	0.70	0.004
5	0.75	0.003
6	0.97	0.96
7	1.0	1.0
8	0.57	0.04

the MAP scores in terms of which ranking produced more relevant results per query.

The highest value between the Google and HighTime rankings for each query is mirrored in both the NDCG and the DCG.

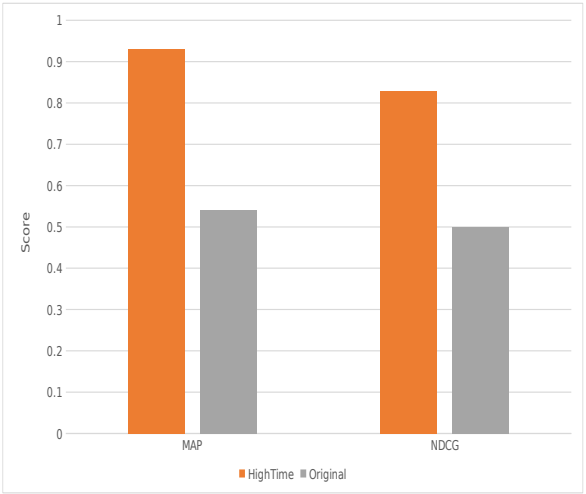


Figure 5: Average MAP and NDCG scores across all queries

Figure 5 shows the average MAP and NDCG scores across all queries for the HighTime and Google rankings. Both MAP and NDCG values for the top 10 HighTime results were higher than those returned by Google.

6.1.1 Determining Statistical Significance. A paired, one-tailed t-test was performed on the data for the MAP and NDCG scores as recommended by Sanderson and Zobel [22] for use in information retrieval. Calculations were performed on the basis of two central hypotheses:

(1) Null hypothesis: The two rankings (produced by Google, and re-ranked with the HighTime algorithm) were equally as good.
(2) Alternative hypothesis: one ranking produced higher scores than the other, or more simply: the two rankings were not equally as good.

As seen in Table 3 and Table 5, there is an increase in the MAP and NDCG scores between the original (Google) and HighTime rank. The increase in the NDCG scores is not significant with p value is equal to 0.053 ie. greater than 0.05 and therefore does not reject null hypothesis. On the other hand, MAP scores are statistically significant with p 0.02 and therefore we reject the null hypothesis. Thus, the ranking produced by HighTime proves better than the default ranking.

7 DISCUSSION

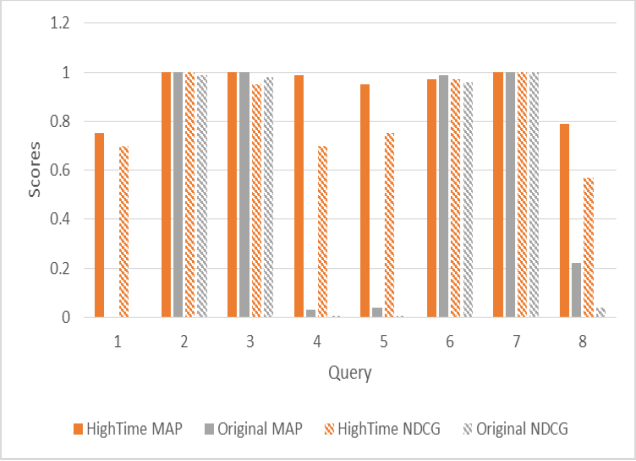


Figure 6: Graph comparing the MAP and the NDCG scores across all queries

For ambiguous queries where the common interpretation of the meaning was most often retrieved by Google (for example with the query "Java", where the order of results prioritized those concerning the programming language), both MAP and NDCG scores for the two rankings were similar. This was observed in queries 2, 3, 6, and 7. MAP and NDCG values of the two rankings for these types of queries had a maximum difference of 0.02 and 0.03 respectively. However, the greatest improvement in relevancy scores could be seen in the examples of queries where the interpretation of the query was less popular for the Google results – for example, in the query "Java" where the interpretation was about the coffee. While Google would return the same results for both interpretations of the query, the HighTime algorithm would re-rank results depending on the time for a topic. This produced results that were tailored to the individual user and personalized based on information in the user profile built through past searches and click-through data. This was observed in queries 1, 4, 5, and 8.

Figure 6 shows a graph of the MAP and NDCG scores across all queries and highlights this principle. Queries 2, 3, 6, and 7 all show similar trends, while queries 1, 4, 5, and 8 show a stark difference that illustrates the effectiveness of HighTime algorithm in removing ambiguity from queries and personalizing results to the user.

This mirrors the investigation done by Sanderson [22], where relevant results for only one interpretation could be returned for 70 percent of ambiguous queries. HighTime improved upon the 4 secondary interpretations of the ambiguous queries, and resulted in average MAP and NDCG scores of 0.93 and 0.83 respectively as compare to 0.54 and 0.50 for Google as shown in Figure 5

8 CONCLUSION AND FUTURE WORK

By using the intent of the user as implicit information in the ambiguous query, HighTime produced results that were more relevant to the user. Results show that the HighTime algorithm re-ranked results in an order that scored a greater measure of relevance than the original order of results produced by Google. Both the MAP and NDCG scores for the HighTime algorithm were greater than those of the Google results, and statistical analysis indicated significance in the experimental results – disproving the null hypothesis that the two rankings would be equally as relevant. The HighTime algorithm worked especially well at removing ambiguity in queries, illustrated by the stark contrast in MAP and NDCG scores between the HighTime and Google results for the secondary interpretations of an ambiguous query.

Future work will include real user testing that would help in refining the algorithm by providing realistic cases and online, real-time feedback. This investigation only looked at comparisons between the performance of HighTime against the original Google results. While this showed that HighTime was able to improve on the original rankings, future studies should look at comparing HighTime against other re-ranking algorithms that have proven to be effective in Web-search personalization.

ACKNOWLEDGEMENT

Thanks go to Tashiv Sewpersad and Jordan Kadish, who worked on related aspects of the project. This research was partially funded by the National Research Foundation of South Africa (Grant numbers: 85470 and 88209) and University of Cape Town. The authors acknowledge that opinions, findings and conclusions or recommendations expressed in this publication are that of the authors, and that the NRF accepts no liability whatsoever in this regard.

REFERENCES

[1] Eugene Agichtein, Eric Brill, and Susan Dumais. 2006. Improving Web Search Ranking by Incorporating User Behavior Information. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '06)*. ACM, New York, NY, USA, 19–26. https://doi.org/10.1145/1148170.1148177

[2] Yossi Azar, Iftah Gamzu, and Xiaoxin Yin. 2009. Multiple intents re-ranking. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*. ACM, Bethesda, MD, USA, 669–678.

[3] Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, and Thomas White. 2005. Evaluating Implicit Measures to Improve Web Search. *ACM Trans. Inf. Syst.* 23, 2 (April 2005), 147–168. https://doi.org/10.1145/1059981.1059982

[4] Carol Jean Godby. 1999. Wordnet: An Electronic Lexical Database. Christiane Fellbaum. *The Library Quarterly* 69, 3 (1999), 406–408. https://doi.org/10.1086/603115

[5] Gregory Grefenstette. 1997. Short query linguistic expansion techniques: Palliating one-word queries by providing intermediate structure to text. In *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology*. Springer, Frascati, Italy, 97–114.

[6] Taher H Haveliwala. 2003. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE transactions on knowledge and data engineering* 15, 4 (2003), 784–796.

[7] Hua Jiang, Yong-Xing Ge, Dan Zuo, and Bing Han. 2008. TIMERANK: A method of improving ranking scores by visited time. In *Machine Learning and Cybernetics, 2008 International Conference on*, Vol. 3. IEEE, Kunming, China, 1654–1657.

[8] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2017. Accurately Interpreting Clickthrough Data As Implicit Feedback. *SIGIR Forum* 51, 1 (Aug. 2017), 4–11. https://doi.org/10.1145/3130332.3130334

[9] Shahid Kamal, Roliana Ibrahim, and Imran Ghani. 2016. Post-search Ambiguous Query Classification Method Based on Contextual and Temporal Information. In *Asian Conference on Intelligent Information and Database Systems*. Springer, Berlin, Heidelberg, 575–583.

[10] Nattiya Kanhabua and Kjetil Nørvåg. 2010. Determining time of queries for re-ranking search results. In *International Conference on Theory and Practice of Digital Libraries*. Springer, Berlin, Heidelberg, 261–272.

[11] Robert Krovetz and W Bruce Croft. 1992. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems (TOIS)* 10, 2 (1992), 115–141.

[12] Sarvesh Kumar, SK Jain, and RM Sharma. 2014. Diversification of web search results using post-retrieval clustering. In *Computer and Communication Technology (ICCCT), 2014 International Conference on*. IEEE, 1–6.

[13] Seung Eun Lee and Dongug Kim. 2013. A Click Model for Time-sensitive Queries. In *Proceedings of the 22Nd International Conference on World Wide Web (WWW '13 Companion)*. ACM, New York, NY, USA, Article 2487859, 2 pages. https://doi.org/10.1145/2487788.2487859

[14] Michael Lesk. 1986. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation (SIGDOC '86)*. ACM, New York, NY, USA, Article 318728, 3 pages. https://doi.org/10.1145/318723.318728

[15] Yizhou Lu, Benyu Zhang, Wensi Xi, Zheng Chen, Yi Liu, Michael R. Lyu, and Wei-ying Ma. 2004. The Powerrank Web Link Analysis Algorithm. In *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers &Amp; Posters (WWW Alt. '04)*. ACM, New York, NY, USA, Article 1013422, 2 pages. https://doi.org/10.1145/1013367.1013422

[16] Thomas Mandl. 2009. Artificial intelligence for information retrieval. In *Encyclopedia of Artificial Intelligence*. IGI Global, 151–156.

[17] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schutze. 2009. An information to information retrieval.

[18] Jivashi Nagar and Hussein Suleman. 2017. Investigating Per-user Time Sensitivity of Search Topics. In *Joint Proceedings of the 1st Workshop on Temporal Dynamics in Digital Libraries (TDDL 2017), the (Meta)-Data Quality Workshop (MDQual 2017) and the Workshop on Modeling Societal Future (Futurity 2017) co-located with 21st International Conference on Theory and Practice of Digital Libraries (TPLD 2017)*, Vol. 2038. CEUR-WS.org, Thessaloniki, Greece. http://ceur-ws.org/Vol-2038/paper2.pdf

[19] Antti Oulasvirta, Janne P. Hukkinen, and Barry Schwartz. 2009. When More is Less: The Paradox of Choice in Search Engine Use. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*. ACM, New York, NY, USA, Article 1572030, 8 pages. https://doi.org/10.1145/1571941.1572030

[20] Soo Young Rieh. 2003. Investigating Web searching behavior in home environments. *Proceedings of the American Society for Information Science and Technology* 40, 1 (2003), 255–264.

[21] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text Mining: Applications and Theory* (2010), 1–20.

[22] Mark Sanderson. 2008. Ambiguous Queries: Test Collections Need More Sense. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*. ACM, New York, NY, USA, 499–506. https://doi.org/10.1145/1390334.1390420

[23] Rodrygo L.T Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting Query Reformulations for Web Search Result Diversification. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*. ACM, New York, NY, USA, Article 1772780, 10 pages. https://doi.org/10.1145/1772690.1772780

[24] Md Shajalal, Md Zia Ullah, Abu Nowshed Chy, and Masaki Aono. 2016. Query subtopic diversification based on cluster ranking and semantic features. In *Advanced Informatics: Concepts, Theory And Application (ICAICTA), 2016 International Conference On*. IEEE, 1–6.

[25] H Sheng, AS Goker, and Daqing He. 2001. Web user search pattern analysis for modelling query topic changes. *Lecture Notes in Computer Science* 2109 (2001).

[26] Ahu Sieg, Bamshad Mobasher, and Robin Burke. 2007. Web Search Personalization with Ontological User Profiles. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management (CIKM '07)*. ACM, New York, NY, USA, Article 1321515, 10 pages. https://doi.org/10.1145/1321440.1321515

[27] Mark D Smucker, James Allan, and Ben Carterette. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM, 623–632.

[28] Ruihua Song, Zhenxiao Luo, Jian-Yun Nie, Yong Yu, and Hsiao-Wuen Hon. 2009. Identification of ambiguous queries in web search. *Information Processing & Management* 45, 2 (2009), 216–229. https://doi.org/10.1016/j.ipm.2008.09.005

[29] Wei Song, Ying Liu, Lizhen Liu, and Hanshi Wang. 2016. EXAMINING PERSONALIZATION HEURISTICS BY TOPICAL ANALYSIS OF QUERY LOG. *INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING, INFORMATION AND CONTROL* 12, 5 (2016), 1745–1760.

[30] Jaime Teevan, Susan T Dumais, and Eric Horvitz. 2005. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 449–456.

[31] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. 2010. Potential for Personalization. *ACM Trans. Comput.-Hum. Interact.* 17, 1, Article 4 (April 2010), 31 pages. https://doi.org/10.1145/1721831.1721835

[32] Md Zia Ullah, Md Shajalal, Abu Nowshed Chy, and Masaki Aono. 2016. Query Subtopic Mining Exploiting Word Embedding for Search Result Diversification. In *Information Retrieval Technology*. Springer, 308–314.

[33] Alexander JAM Van Deursen and Jan AGM Van Dijk. 2009. Using the Internet: Skill related problems in users' online behavior. *Interacting with computers* 21, 5-6 (2009), 393–402.

[34] Maksims Volkovs. 2015. Context models for web search personalization. *arXiv preprint arXiv:1502.00527* (2015).

[35] Thanh Tien Vu, Alistair Willis, and Dawei Song. 2015. Modelling time-aware search tasks for search personalisation. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 131–132.

[36] Deng Yi, Yin Zhang, and Baogang Wei. 2016. Query Subtopic Mining via Subtractive Initialization of Non-negative Sparse Latent Semantic Analysis. *J. Inf. Sci. Eng.* 32, 5 (2016), 1161–1181.