Multiple Clicks Model for Web Search of Multi-clickable Documents

Léa Laporte[1,3], Sébastien Déjean[2] and Josiane Mothe[1,4]
[1]*Institut de Recherche en Informatique de Toulouse, UMR 5505, Université de Toulouse, 118 Route de Narbonne, 31062 Toulouse Cedex 9, France*
[2]*Institut de Mathématiques de Toulouse, UMR 5219, Université de Toulouse, 118 Route de Narbonne, 31062 Toulouse Cedex 9, France*
[3]*Nomao SA, 1 Avenue Jean Rieux, 31500 Toulouse, France*
[4]*IUFM Midi-Pyrnes, 56 Avenue de l'URSS, 31078 Toulouse cedex 4, France*
{*laporte, mothe*}*@irit.fr, lea@nomao.com, sebastien.dejean@math.univ-toulouse.fr*

Keywords: Information retrieval, Web search, Clickthrough information, Evaluation, Document relevance model.

Abstract: This paper presents a novel document relevance model based on clickthrough information. Compared to the models from the literature we consider the case when documents can be clicked several times in a given search session. This case occurs more and more frequently, specifically for multi-clickable documents such as maps in location search enginess. Considering a real system query log, we evaluate our model and show that SVM can learn with fewer errors and with better MAP when the various types of clicks are considered in the model.

1 INTRODUCTION

Information Retrieval (IR) systems aim at retrieving documents that answer information needs that users expressed in queries. Evaluation is a core component in IR. The dominant evaluation framework in the domain is the one set in the Cranfield project (Cleverdon et al., 1966), which relies on performance measures to be calculated using a test collection composed of a collection of documents, a collection of queries and a collection of relevance assessments collected manually. Whereas full relevance judgment has been possible in Cranfield because of the relatively small size of the document collection (1400 documents) and query set used, more recent evaluation projects such as NIST[1], CLEF[2] or INEX[3] rely on the pooling method in which relevance judgments are collected on the set of documents selected by the various participating systems.

Even if it allows researchers to get gold standard collections to evaluate various IR methods, the pooling method has some drawbacks (Harman, 2010). One of them is that some relevant documents are not in pools; however detecting relevant documents that could not be retrieved by any system can hardly be circumvented in large scale evaluation. A probably more critical drawback is that the user model is somehow ignored by this relevance model: the process of the formation of information need and of relevance assessment requires realism (Borlund, 2003). A document should be considered as relevant if the user uses it in some ways and if it answers a real users' need. When building adhoc collections, users' information needs are solicited rather than corresponding to real ones. Relevance models based on logs and clickthrough information aim at answering these issues while providing large scale evaluation collections (Joachims, 2002; Dupret et al., 2007). In addition, they provide a non-intrusive way of collecting information regardless users' actions; the latter being not well-accepted by users.

However, deriving reliable judgments from implicit feedback is not trivial. It is well established that clicks cannot be considered on an absolute scale but rather than clicked results are better than skipped results (Yue et al., 2010). Various models have been developed to evaluate systems by using clickthrough information (Chapelle and Zhang, 2009; Craswell et al., 2008; Guo et al., 2009; He et al., 2011).

Existing models consider that several results can be clicked in a search session. However, they do not take into account the fact that a given reference can be clicked through several times in a single session (Laporte et al., 2012). For example, a user can choose a reference, go back to the initial list of references and

[1]http://trec.nist.gov/
[2]www.clef2013.org
[3]https://inex.mmci.uni-saarland.de/

choose another one he has clicked through just before. The multiple click feature on a given reference can also occur in the case of some types of documents such as maps that provide various information (several locations for example), which is more and more common. Other examples are on digital libraries that gives access to collections through a list of results on which the user can either click on the title, view the document or have access to a record of information such as its author and his/her other publications, the publication date or the editors. Each action is a clue on the document relevance.

In this paper we present a new multiple-click model in web search. As opposed to other models we consider multiple clicks of the same document in a single session as a relevance clue. The second contribution of this paper is an experimental study based on a data set from a geo-referenced information search engine. It demonstrates the usefulness of our model in learning the function used to rank the documents to be retrieved according to a user's query.

The rest of the paper is organized as follows. We first present some related works in Section 2. In Section 3, we present our multiple-click model. In Section 4, we report experimental studies. Section 5 discusses the results and concludes this paper.

2 RELATED WORKS

Building gold standard collections is a key issue in IR. Manual building is a long and demanding process. Clickthrough information conveys somehow user's interest and for this reason can be considered as users' implicit relevance judgments.

(Joachims, 2002) and (Joachims et al., 2005) proposed methods to extract users' preferences based on click information. The *SkipAbove* strategy considers that the last clicked document should be preferred to any document retrieved at a higher rank. (Radlinski and Joachims, 2005) expand this rule in the case of query sessions. Other works focus on the so called position bias. It refers to the fact that users have the tendency to choose from the first retrieved documents. The position model proposed in (Craswell et al., 2008) considers that documents have different click probabilities and that this probability should decrease with the document rank. This model considers that a single document is clicked. The cascade model is different since it considers that users view results from top to bottom and leave as soon as they see a relevant document (Craswell et al., 2008). Both models make a strong assumption that a single document is clicked in a given search. The dependent click model proposed in (Guo et al., 2009) generalizes the cascade model and takes into account search sessions with multiple clicks. The model hypothesizes that the user browses the retrieved list from top to bottom, and can go back to the list after he has selected one item. The document relevance is then computed overall, considering the various users. In (Guo et al., 2009), the authors show that the document relevance can be written as the number of clicks on that document on the session set divided by the number of sessions that retrieved the document.

The dynamic bayesian network click model (Chapelle and Zhang, 2009) also generalizes the cascade model to multiple click sessions. They introduce two types of relevance: the perceived relevance and the actual relevance. The latter relevance corresponds to the fact that the user is satisfied by the results of the search: when the user's need is satisfied, he stops the session; otherwise, he goes back to the search result list. The global relevance is calculated as the product of the perceived relevance and the actual relevance; it corresponds to the final relevance judgment. In (Liu et al., 2010), a Bayesian model is used with multiple click sessions. Rather than predicting a relevance score, it predicts the probability that a document is preferred to another. This approach is interesting for systems that use pairwise algorithms when ranking. In (Yue et al., 2010) suggest an alternative way of scoring differently clicks depending on when they occur in the search process. These scores are used to compute the document relevance. In addition, this work as well as the one in (He et al., 2011) optimizes the size of the data set needed for evaluation.

3 MULTIPLE-CLICK MODEL

The proposed model focuses on the case when a document can receive multiple clicks in a single session.

Following previous related work in the domain, we aim at defining a relevance score based on users' clicks. We hypothesize that a click is an implicit judgment of document relevance. We also consider the case when a given document can be clicked several times in a given session, mainly because it provides several links to various data (phone number, pop-up information, addresses...). We thus hypothesize that the more a document is clicked through, the more relevant it is. Relevance can be computed according to the number of clicks it received. This approach corresponds to the simple multi-click model. However this approach considers that any action on the document, any click is equivalent and depicts a similar interest from the user. On the contrary, we think that the vari-

ous types of actions can be considered differently. For example, the fact that a user clicks on the phone number item on a document should be weighted differently than other clickthroughs. When clicking on the phone number item for example, which automatically shows the current phone number and eventually connect the user to a phone service, the user expresses the fact the document is of sufficient interest to call the phone owner. Intuitively, different actions or clicks on different items from a clickable document imply different levels of interest for the user. This intuition leads us to complete the model considering not only the number of clicks but also the various natures of the links that can be clicked through. We finally define the relevance score as follows:

$$r_{d,q,u} = \sum_i \alpha_i c_{i,d,q,u} \quad (1)$$

where $r_{d,q,u}$ is the relevance of document d for the query q and user u, $c_{i,d,q,u}$ is the number of clicks of type i on the document d for the query q and user u and α_i is a coefficient that traduce the importance of the click of type i.

The α_i parameter is worth studying since it indicates the importance of each type of click in the model. When $\alpha_i = 1$ whatever i is, the model corresponds to the simple model we start with. This model is a good start to evaluate ranking functions in the case of multi-clickable documents. This evaluation aims at demonstrating that defining the query-document relevance score based on this model allows learning a ranking function in an efficient and effective way. We also study in the rest of the paper the influence of the various types of clicks on learning to rank documents. We aim at defining an order of importance of the various types of clicks in order to learn the best ranking function.

4 EXPERIMENTAL STUDIES

4.1 Preliminary analysis

This first analysis is based on a limited log file from Nomao search engine[4] and aims at getting information on the data. Nomao allows users to search for locations that are recommended according to the users profiles based on their tastes, location and social network. Results are presented under the form of documents in which several items can be clicked through by users and under the form of geographic maps.

We extract several elements from the logs: the query identifier and content, the first 20 documents corresponding to the system answer, the number and the type of clicks for each user-query-document set.

We then extract query-document pairs to be associated to a relevance score. We identify five types of clicks: on the document title (α_{title}), booking button ($\alpha_{booking}$), phone number (α_{phone}), web site (α_{URL}) and the result from the geographic map (α_{map}).

In this first analysis, we keep solely queries that are associated at least with two retrieved documents and for which an item has been clicked through. We extracted 14700 user-query-document sets for which the number and the types of clicks are known. This collection corresponds to 2014 distinct queries, 1745 different users, and 14343 documents.

The average rank on the first page of results is 3.54: users click on the third or fourth document from the retrieved list. However, the median of the click rank is 1: at least half of the clicks concern the first retrieved document. This result may illustrate the position bias (users click through the first retrieved document even if not relevant). It can also be the results of unequally performance on queries (some queries being easier for the system than others).

We found out that the number of clicks of the various types is different. Clicks on title, phone and map are the most frequent (44%, 28%, and 24% respectively). Clicks on the web site (3%) and on the booking button (1%) are far less numerous. For some queries, users only use a single type of clicks. For example, 37% of the queries imply title clickthrough action only. Queries associated with a single type of click correspond to 22% for phone only, 12% for map only, 1% web site only, and 0.6% for booking only.

Finally, the number of retrieved documents is different from one query to another. 38% of the queries lead to 20 retrieved documents whereas 26% retrieves 3 or less documents. This distribution has an impact on the results over the evaluation measures. In this study we consider the precision at rank k ($P@k$) which is defined as the number of relevant documents the system retrieves when considering the first k retrieved documents. We also use the mean average precision (MAP) over queries.

4.2 Experimental setup

We evaluate the relevance document model considering the learning to rank task.

Learning to rank in IR aims at automatically deciding the best ranking function to order the retrieved documents. Considering a query, learning to rank aims at ranking documents according to document relevance. Ranking is based on various features associated to queries, query-document similarities and

[4] http://www.nomao.com/

documents. Ranking functions are usually learned on training data consisting of query-document pairs for which relevance is known. The trained functions are then used to rank documents on new queries.

Learning to rank algorithms can be grouped into three types which differ on the way they consider learning (Liu, 2011). The pointwise approach considers documents independently. To each document is associated a relevance score for a given query. The learning process either uses regression (Cossock and Zhang, 2006) or classification (Nallapati, 2004). Inputs of the pairwise approach are document pairs to which preferences are associated to. For a given query, a preference value of 1 for pair (d_1, d_2) implies that d_1 is more relevant than d_2 whereas the reverse preference -1 means that d_2 should be ranked higher than d_1. The learning problem can be reduced to a classification problem. Various classifiers, including Support Vector Machine (Joachims, 2002; Chapelle and Zhang, 2009), Neural Networks (Burges et al., 2005) or Boosting (Freund et al., 2003), have been proposed to solve this problem. Finally, the listwise approach uses an ordered list of documents as input. The ranking function minimizes the distance between the obtained ranked list and the reference ranked list (Cao et al., 2007) or optimizes an information retrieval performance measure (Yue et al., 2007).

In the following experiments, we use the same learning algorithm for which we consider 145 features corresponding to query, query-document matching, document and user features. We analyze the impact of the scores on the quality of the ranking. We conduct two main experiments to evaluate our approach and to determine the impact of each type of clicks. In the first experiment, we evaluate our model when all the clicks are weighted to 1. In the second experiment, we consider different scores for which all types of clicks received the same weight except for one type which weight is set to 0, so that we can evaluate the impact of each type of clicks.

4.3 RankSVM

We use RankSVM for learning to rank. RankSVM implements a pairwise ranking SVM as proposed in (Joachims, 2002). It predicts the relevance judgments by minimizing the following function (Liu, 2011):

$$\min \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}\sum_{u,v,y^{(i)}} \xi^{(i)} \tag{2}$$

under the constraints

$$\begin{cases} w\top(x_u^{(i)} - x_v^{(i)}) \geq 1 - \xi_{u,v}^{(i)} \text{ if } y_{u,v}^{(i)} = 1 \\ \text{and } \xi_{u,v}^{(i)} \geq 0 \forall i = 1, \ldots, n \end{cases} \tag{3}$$

where n is the number of queries of the training set, $(x_u^{(i)} - x_v^{(i)})$ the pair of documents u and v represented in the feature space and associated with the query i, $y_{u,v}^{(i)}$is the preference judgment for the document pair, and $\xi_{u,v}^{(i)}$ corresponds to the error for the non-linearly separable SVM. C allows the control of the error rate. It is chosen by cross validation.

In order to evaluate the robustness of the model, we simulated 5 data collections from the initial data set presented in section 4.1, each one containing training data and test data. To build one collection, we randomly associate the queries from the initial set either to the training set or to the test set. The size of the training set is 80% of the initial set so that the test set contains 20% of the initial queries. We build 5 different collections this way and use them in experiments. The C parameter is set by 10-folds cross validation considering the following values $\{0.001, \ldots, 0.009, 0.01, \ldots, 1, 1.1, \ldots, 1.3\}$.

Regarding effectiveness, we obtained high MAP values (from 0.78 to 0.80) during these preliminary experiments. Comparing to MAP performance of RankSVM on LETOR datasets[5], these values can seem high. This behaviour is due to the amount of queries that have a few number of retrieved documents associated to. Indeed, for queries that have only two retrieved documents associated to, the average precision is either 1 or 0.5. Taking into account these queries for the MAP computation will lead to high MAP values. However, this does not affect the reliability of the measure that can be used to compared scores between various versions of the process.

4.4 Results

4.4.1 Equal contribution of the clicks

In this experiment, we analyze the equally contribution of the various types of clicks and compare the results with a random binary relevance score. To do so, in one hand we randomly associate a relevance score to the document (0 for non-relevant; 1 for relevant); in the other hand in formula 1 that describes our model, we consider $\alpha_i = 1$ whatever i. This is done for each user-query-document set. The system learns the ranking function using the score defined these ways. We then compare the processing time of the two models. A higher processing time indicates a greater difficulty to learn the function and thus a less powerful relevance score. We also compare the prediction error as well as MAP which are effectiveness indicators.

[5] MAP varies from 0.22 to 0.74 on test set depending on the dataset

The results show that whatever the measure, we obtain better results using the multiple click method than when considering the random relevance score. First, the time needed to process the cross validation decreases about 15000 seconds (random) to 10000 seconds (multiple clicks). In the same way, we found a prediction error on the test set equal to 82.7% for the random model and equal to 38.6% for the multiple click model. In the case of random relevance scores, the algorithm fails to learn the ranking function. Finally, MAP is about 0.65 using the random model and is about 0.78 using the multiple click model. This experiment shows that using the number of times a document has been clicked through makes sense.

4.4.2 Weighting the various types of clicks

This version of the multiple click model we promote hypothesizes that each type of action (click type) has a different impact on relevance clue. Moreover, we think that the clicks could be ordered according to their importance in their contribution on the relevance score. In this experiment set, we consider five scores for which we successively set the contribution of one type of clicks to zero. We then analyze the impact on the MAP and prediction error. When comparing two runs, lower values of MAP associated with higher values of prediction error show that the corresponding algorithm does not learn well the ranking function so that the corresponding scores are less performing.

The MAP values we obtained are presented in table 1. When considering the MAP, the first test set has a different behavior than the others and lower MAP. Results in terms of prediction error are similar, so we only present and comment the MAP values.

When considering the test set, the MAP is the lower for $\alpha_{booking} = 0$, when the booking button is not considered. The learning algorithm does not predict the relevance in an accurate way in that case, which means booking clicks play an important role in the relevance model. Intuitively, this result makes sense. When a user clicks on the booking link, his is likely satisfied by the system answer. On the contrary, the result we obtained regarding $\alpha_{phone} = 0$ is quite unlikely. On a practical view, when a user clicks on the phone link it is supposed to mean that the user wants to see the phone number and is likely to contact the corresponding location. The booking and phone links looks quite similar in an information usage point of view, so we expected to get similar results for both $\alpha_{booking} = 0$ and $\alpha_{phone} = 0$. However, the MAP is the highest when clicks on the phone links are not considered. Phone clicks are noisy for the multiple click model. As the coverage of MAP in that case is quite large, the results seem quite collection dependent. We also observe that the MAP is lower when the clicks on the map are not taken into account. Those clicks are also important in the multiple click model.

5 DISCUSSIONS AND CONCLUSIONS

In this paper, we propose a novel document relevance model based on clickthrough information. While existing models consider that either a single document is clicked in a search session or several documents can be clicked but a given document only once, we consider the case when documents can be clicked several times in a given session. This case occurs more and more frequently, especially for multi-clickable documents such as maps in location search engines.

We evaluate our model on a real system query log and show that SVM can learn with fewer errors and with better MAP when the various types of clicks are weighted differently in the model. Future works will focus on refining the weights that should be associated to each type of clicks. The results we obtained suggest that booking and map clicks should be weighted more than title and web site links. We will investigate the behaviour of phone clicks. We suspect that the bad results they induce is due to the fact users are redirected to overtaxed services that when clicking phone links. The users may then prefer to go back to the result list to find other free information.

Further investigations are currently conducted on a larger queries log from the Nomao search engine. Massive analyses are planned by using the resources of the supercomputer Hyperion from the scientific group CalMip[6]. Preliminary experiments confirm the impact on the MAP of queries with few retrieved documents. Indeed, when considering the score $\alpha_i = 1$ whatever i, the MAP increases from 0.30 to 0.60 when we include queries with only two documents retrieved. The analysis of clicks distribution shows an increase of the number of clicks on the phone link. This may be explained by changes on phone link design. Further analysis are planned to evaluate the impact of this new design on the users' behavior.

ACKNOWLEDGEMENTS

We would like to thank the Région Midi-Pyrénées, the FREMIT federation and CalMip for their support.

[6] www.calmip.cict.fr

Table 1: MAP on the various test sets

Test set	Score					
	$\alpha_i = 1 \forall i$	$\alpha_{booking}$	α_{phone}	α_{URL}	α_{map}	α_{title}
1	0.75	0.75	0.73	0.76	0.74	0.70
2	0.81	0.73	0.77	0.79	0.77	0.76
3	0.78	0.77	0.79	0.80	0.76	0.79
4	0.75	0.75	0.81	0.76	0.73	0.77
5	0.79	0.74	0.81	0.76	0 .75	0.79
Average	0.78	**0.75**	**0.78**	0.77	**0.75**	0.76
Coverage	0.06	0.04	0.08	0.04	0.04	0.09

REFERENCES

Borlund, P. (2003). The iir evaluation model: a framework for evaluation of interactive information retrieval systems. *Information research*, 8(3):8–3.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 89–96.

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., and Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 129–136.

Chapelle, O. and Zhang, Y. (2009). A dynamic bayesian network click model for web search ranking. In *Proceedings of 18th International Conference on World Wide Web*, WWW'09, pages 1–10.

Cleverdon, C. W., Mills, J., and Keen, M. (1966). Factors determining the performance of indexing systems.

Cossock, D. and Zhang, T. (2006). Subset ranking using regression. In *Proceedings of the 19th annual conference on Learning Theory*, COLT'06, pages 605–619. Springer-Verlag.

Craswell, N., Zoeter, O., Taylor, M., and Ramsey, B. (2008). An experimental comparison of click position-bias models. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 87–94.

Dupret, G., Murdock, V., and Piwowarski, B. (2007). Web search engine evaluation using clickthrough data and a user model. In *WWW2007 workshop Query Log Analysis: Social and Technological Challenges*.

Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969.

Guo, F., Liu, C., and Wang, Y. M. (2009). Efficient multiple-click models in web search. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 124–131.

Harman, D. (2010). Is the cranfield paradigm outdated. In *Proc. 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1.

He, J., Zhao, W. X., Shu, B., Li, X., and Yan, H. (2011). Efficiently collecting relevance information from clickthroughs for web retrieval system evaluation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 275–284.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 133–142.

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, pages 154–161.

Laporte, L., Candillier, L., Déjean, S., and Mothe, J. (2012). Évaluation de la pertinence dans les moteurs de recherche géoréférencés. In *Actes du 30me Congrés INFORSID*, pages 281–298.

Liu, C., Guo, F., and Faloutsos, C. (2010). Bayesian browsing model: Exact inference of document relevance from petabyte-scale data. *ACM Transaction on Knowledge Discovery Data*, 4(4):19:1–19:26.

Liu, T.-Y. (2011). *Learning to rank for information retrieval*. Springer.

Nallapati, R. (2004). Discriminative models for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 64–71.

Radlinski, F. and Joachims, T. (2005). Query chains: learning to rank from implicit feedback. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 239–248.

Yue, Y., Finley, T., Radlinski, F., and Joachims, T. (2007). A support vector method for optimizing average precision. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 271–278.

Yue, Y., Gao, Y., Chapelle, O., Zhang, Y., and Joachims, T. (2010). Learning more powerful test statistics for click-based retrieval evaluation. In *Proceedings of the 33rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'10, pages 507–514.