Mailbox-Based vs. Log-Based Query Completion for Mail Search

Michal Horovitz*
Technion, Israel
michalho@cs.technion.ac.il

Liane Lewin-Eytan
Yahoo Research, Israel
liane@yahoo-inc.com

Alex Libov
Yahoo Research, Israel
alibov@yahoo-inc.com

Yoelle Maarek
Yahoo Research, Israel
yoelle@yahoo.com

Ariel Raviv
Yahoo Research, Israel
arielr@yahoo-inc.com

ABSTRACT

Recent research studies on mail search have shown that the longer the query, the better the quality of results, yet a majority of mail queries remain very short and searchers struggle with formulating queries. A known mechanism to assist users in this task is query auto-completion, which has been highly successful in Web search, where it leverages huge logs of queries issued by hundreds of millions of users. This approach cannot be applied directly to mail search as personal query logs are small, mailboxes are not shared and other users' queries are not necessarily generalizable to all. We therefore propose here to leverage the mailbox content in order to generate suggestions, taking advantage of mail-specific features. We then compare this approach to a recent study that augments an individual user's mail search history with query logs from "similar users", where the similarity is driven by demographics. Finally we show how combining both types of approaches allows for better suggestions quality but also increases the chance that the desired message be retrieved. We validate our claims via a manual qualitative evaluation and large scale quantitative experiments conducted on the query log of Yahoo Mail.

CCS CONCEPTS

•Information systems →Information retrieval;

KEYWORDS

Mail search, Query Auto Completion, Query Suggestion

1 INTRODUCTION

Web mail search has recently attracted more attention from the research community [1, 7, 10, 13]. One recent study by Carmel et. al. in particular [6] highlighted the fact that users struggle more with formulating queries in mail than in Web search, with an average query length of 1.5 as opposed to 3 terms. The authors also show that, like in other domains, more specific queries lead to higher quality results, which motivates the need for better query assistance. The most dominant approach for query assistance today is query auto-completion that has been highly successful in Web search [4]. However, in the context of mail search, query completion presents two key challenges: First, the personal query log (i.e., a single user's search history), which is the primary source of auto-completion, is sparse; second, as argued in [1], messages are not shared across and queries, which are most often personal, might not generalize well to other users. One solution to address the first challenge is to increase the size of query logs, as done in [6], where the authors leverage the logs of users with similar demographic characteristics. This however does not address our second challenge. Even if some mass-senders send similar machine-generated messages to similar users, the key information such as a trip destination or a purchased product cannot be generalized across users. This leads us to consider leveraging additional sources such as mailbox content. Leveraging content has been used in the past for improving logs suggestions [9, 12]. In the enterprise domain, [11] took advantage of additional sources such as structured catalogs to increase coverage. Closer to our work Bhatia et al.[3], proposed to generate suggestions directly from the corpus in absence of query logs, yet, they did not specifically tailor their approach to mail.

We propose here to study the benefits of leveraging mailbox contents for mail query auto completion. More specifically, we propose to generate suggestion candidates by extracting N-grams directly from the user's messages, and rank them using a model specifically-tailored to mail. We go beyond simple frequency counts and consider a wide range of mail-specific features at different levels of granularity, from mailbox-level to field-level features. For example, we associate with the generated suggestion, attributes from the message it originates from, such the message freshness, the folder it belongs to, or the actions that were performed over it, such as read, reply, etc. We compare the quality of these message-based suggestions to those generated by query logs, using the method proposed by [6] as our baseline. In addition, we present a combined approach, where suggestions originate from both query logs and mailbox content. We apply a comprehensive ranking model trained over the whole set of features. Leveraging mailbox content allows us to validate suggestion candidates derived from other users' logs, thus addressing challenge (2). We evaluate our approach in a set of experiments we conducted on Yahoo Mail.

2 MAIL QUERY AUTO COMPLETION

In the most common settings, the query completion mechanism takes as input a short string of characters entered by the user and returns a ranked list of full queries, referred as completions or more generally *suggestions* [4]. The ranking of suggestions can be done

*This work was conducted during Michal's internship at Yahoo Research.

SIGIR'17, August 7–11, 2017, Shinjuku, Tokyo, Japan.

DOI: http://dx.doi.org/10.1145/3077136.3080683

via various methods as in other information retrieval tasks, from occurrence frequency considerations to learning-to-rank methods, which we chose to adopt here as detailed later on. We detail below our method for generating suggestions from the mailbox content, as well as a combined approach where suggestions are generated from both mailbox content and query logs.

2.1 Mailbox-Based Suggestions

We first generate from the user's mailbox a list of terms and bigrams. As mail queries are typically short, we did not experiment with higher-order N-grams, which we might consider in future work. Note that as we extract bigrams, we use the same approach adopted by [3], and "jump over" stop words. So for the text *confirmation of order*, instead of generating two bigrams, *confirmation of* and *of order*, none of which is a complete phrase, we generate one bigram, i.e. *confirmation of order*. As a result none of our suggestions start or end with a stop word. We then rank candidates, using a set of features, listed below according to their level of granularity.

Mailbox-Level Features: We propose first to reflect the importance of a given suggestion candidate by considering its frequency of occurrence in a given mailbox as compared to the entire mail domain. We thus use tf-idf score, where the mailbox is unified into a single document and the corpus is the collection of all mailboxes in a given geography (e.g., the US). Formally,

$$tf\,(c, M) = \log\left(1 + \frac{freq\,(c, M)}{\sum_{c' \in C} freq\,(c', M)}\right), \tag{1}$$

where $freq\,(c, M)$ denotes the raw frequency of candidate c in mailbox M, and C is the set of all the candidates from M. Additionally, for a candidate c we define

$$idf\,(c) = \log \frac{|\mathcal{M}|}{|\{m \in \mathcal{M} : c \in m\}|}, \tag{2}$$

where $\mathcal{M}$ is the set of all the messages in our corpus.

The tf-idf score is computed separately for unigrams and bigrams, then all scores are added as features to our model.

Message-Level Features: Our assumption here is that suggestions originating from a less important message (e.g. that has never been read, which is quite common in Web mail) should be demoted.

We use several attributes of a message in order to compute its importance. The first type of attributes are actions performed over a message (e.g. read, flag, forward, etc.), which have been shown to be beneficial in mail relevance [5]. In addition, we consider also whether the message was received or sent, as well as its parent folder. Finally, we consider the message's recency, and use a variant of Equation (1), where we penalize candidates that appear mostly in very old messages. Formally, we replace $freq(c, m)$ in Equation 1 with a time-sensitive function $freq^*(c, m)$ defined as

$$freq^*\,(c, m) = e^{-t_m} \cdot w_m \cdot freq\,(c, m) \tag{3}$$

where t_m denotes the message's age, and w_m is the importance score of message m, which is a function of the attributes mentioned above, automatically learned by our learning-to-rank framework.

Field-Level Features: Message fields (subject, from/to, body, etc.) play an important role in mail ranking [5]. Consequently, we compute the frequency and idf scores for each field. In addition, we account for message decay as per Equation 3, thus giving the follow field-specific computations for tf and idf scores:

$$tf\,(c, f, M) = \log\left(1 + \frac{freq^*\,(c, f, M)}{\sum_{c' \in C} freq^*\,(c', f, M)}\right), \tag{4}$$

and

$$idf\,(c, f) = \log \frac{|\mathcal{M}|}{|\{m \in \mathcal{M} : c \in m_f\}|}, \tag{5}$$

where $freq\,(c, f, M)$ denotes the frequency of candidate c in field f in mailbox M, and m_f is the value of field f in messages m.

We automatically learn the field importance score. Specifically, we consider five fields: *from, to, cc, subject*, and *attachment name*.

Ranking Algorithm: Given the features defined above, we apply a learning-to-rank approach on the suggestions. We use AROW [8], an online variant of SVMRank, which learns a linear weight vector through pairwise comparisons between the relevant candidate and other top-ranked candidates for each query. We tune the learning parameters through standard training, validation and testing on separate sets, where each set includes queries corresponding to a different time-frame. This way, we guarantee that the sets do not overlap and that we remain close to real-world settings.

2.2 Combining Mailbox and Query Log

A major drawback of using content rather than queries as source of suggestions is the well-known query-document mismatch [2]. However, given that at this stage we consider only single terms and bigrams, we reduce the risk of suggestions looking unnatural. In a combined approach, we hope for the better candidates to appear both in the user's mailbox and query logs, and thus be both relevant to the mailbox and look like a potential query. One immediate benefit of using mailboxes in addition to query logs, is to validate suggestions originating from other users' query logs so as not to lead to empty results.

We first consider an *offline validation* method, that verifies that each term that appears in a log-based query suggestion does appear in the user's mailbox, using an offline generated set of all the terms appearing in the mailbox. The limitation of this approach is that a multi-term suggestions with terms appearing in different messages will be validated without necessarily leading to an existing message. We still propose to compare its benefits to the safer *online validation* method, where we validate each of the top candidates using the search index and remove candidates that lead to an empty result list. This *online validation* method, while clearly preventing empty results, presents an obvious overhead in terms of latency.

The actual combination model is done by computing for each candidate a score that combines features from both the query logs and mailbox sources. Formally,

$$S_{comb}(c) = \lambda \cdot S_{content}(c) + (1 - \lambda) \cdot S_{log}(c) \tag{6}$$

where $S_{source}(c)$ denotes the score of candidate c according to *source* and λ is a parameter that is automatically tuned by our framework, which is linear. Two main factors account for the candidate score: (1) its prevalence in important messages with respect to the user's mailbox (2) its likelihood as a query, with respect to the user's past queries as well as the joint query log of the entire user population.

3 EXPERIMENTS

3.1 Dataset

Our dataset consist of a sample of 23.7M queries issued by 4.4M unique users in March-July 2016 in the US traffic of Yahoo Mail. We use it to calculate the frequency distribution of each suggestion, in a joint query log across all users in our dataset, as well as across different demographic sectors, as in [6]. In addition, we sampled 1.8K users who volunteered their data, and processed their mailbox content and personal search history over 2016. Overall, we collected 38M messages, with an average of 21K messages per user. One limitation of learning methods in mail search is that only the owner of the mailbox is capable of generating queries adapted to the mailbox content and selecting the desired result. We therefore followed [14] and employed a method for automatically generating training labels. First, we sampled from our dataset a set of queries followed by a clicked result and generated for each query a set of prefixes that consist of the $1, 2, 3, 4$ first characters and the first term of the query. For each prefix, we obtained all matching query candidates using an auto-completion trie structure over the entire data. Next, for each prefix, we marked as relevant the suggestion that was identical to the query that was issued by the user. We considered only queries that were followed by a click on a result in order to avoid low-quality queries. We refer to this evaluation method as *query-based* evaluation.

The query-based evaluation approach is rather strict, as we consider as irrelevant suggestions that the user might have selected if s/he had been exposed to them. In order to account for this limitation, we evaluate our models in an additional setting. We follow the methodology of [12] and consider a suggestion candidate to be useful if it leads to a search result page, where the position of the clicked message is equal or higher than its position in the search result page of the original query. We refer to this evaluation method as *result-based* evaluation. In the remainder of this section, we primarily use the *query-based* evaluation method, then demonstrate its correlation with the *result-based* evaluation.

3.2 Mailbox-Based Suggestions

We measured the quality of our mailbox-based suggester with respect to different feature sets, using the same metrics as [6], namely Mean Reciprocal Rank (MRR) of the relevant suggestion and *success@k*, with $k = 5$ as mail users are typically presented with a small number of suggestions. Our test set is composed of 20K examples for each prefix length, according to our query-based evaluation method.

Results are presented in Table 1. As expected, the performance of the suggester varies with prefix length. When adding message-level features to unified-mailbox features, we note an increase in MRR of 90% for 1 character, vs. 3.5% for 4 characters. Finally, we observe an additional increase when we consider the message fields, which sums up to an increase of 111.8% for one character and of 4.3% for four characters. As the prefix length grows, the candidate list gets shorter, and there is less room for improvement. In terms of *Success*@5, as before, we observe similar improvement. We retain the Field-based features as our most successful Mailbox-based suggestion method and refer to it as the "Mailbox" method in the following experiments. Note that all reported gains, noted as (+xx%) in Table 1 and the following ones, are statistically significant, as validated by a two-tailed paired t-test ($p < 0.05$).

	Prefix	Mailbox-Level	Message-Level	Field-Level
MRR	1 chars	0.066	0.126 (+90.0%)	0.140 (+111.8%)
	2 chars	0.225	0.309 (+37.0%)	0.326 (+44.7%)
	3 chars	0.419	0.463 (+10.4%)	0.469 (+11.9%)
	4 chars	0.505	0.523 (+3.5%)	0.527 (+4.3%)
	1 term	0.306	0.321 (+4.9%)	0.320 (+4.7%)
Success@5	1 chars	0.089	0.187 (+111.1%)	0.206 (+132.5%)
	2 chars	0.321	0.427 (+33.2%)	0.450 (+40.5%)
	3 chars	0.557	0.584 (+4.8%)	0.588 (+5.5%)
	4 chars	0.621	0.635 (+2.3%)	0.638 (+2.7%)
	1 term	0.364	0.373 (+2.4%)	0.373 (+2.4%)

Table 1: Effectiveness of various features for different prefix lengths. Gains are compared to the $Mailbox - Level$ column.

3.3 Combining Mailbox and Query Log

We use as baseline the suggestion mechanism proposed by [6], which relies on personal and global query logs. The baseline scores at various prefix lengths are shown in the "Logs" column of Table 2 for the query-based evaluation method, and of Table 4 for the results-based evaluation method. We first evaluate the effect of our offline validation method by comparing its MRR and Success@5 scores to our baseline as well as to the more costly online validation. The validation techniques leads to an MRR improvement of 4.1% for 3 chars prefix as compared to 4.6% for the online evaluation. The offline techniques is quite close to the online technique for 1-4 chars, and underperforms only for 1-term prefixes with an improvement of 0.4% as compared to 5.5% for online. This can be attributed to fact that for 1-term prefix the collected queries and candidates would be strictly longer than one term, leading to a difference in the performance of the two techniques as explained earlier. Conversely, for prefixes of only few characters, the gap is much smaller, as most of the query candidates are one term long, corresponding to their distribution in the general query log.

Given that the latency of online validation will be in most cases prohibitive in the case of a large scale mail service like Yahoo Mail, these results are quite encouraging, allowing us to use a cheap offline validation method to reduce the risk of empty results sets. Thus, for further experiments, we use the offline method in order to evaluate our combined model.

We now incorporate mailbox-based suggestions and compute a ranking over all candidates using features from both sources. Table 2 compares the quality of the obtained suggestions to our Logs baseline and to the previous Mailbox method, using the same query-evaluation method. First, we note that the mailbox-based suggester outperforms the log-based one with an increase of between 13.3% and 34.2% in MRR in all settings except for single character prefixes. We assume that in such scenarios, where the number of possible candidates is tremendous, there is more value in a pure log-based approach, which promotes the most useful candidates according to the search patterns of a vast number of users.

As detailed in Table 2, the combined approach performs best, with a significant increase in MRR and Success@5 for all prefix lengths. This demonstrates the value of both sources. On one hand, the personal mailbox content ensures the relevance of suggestions

	Prefix	Logs	Mailbox	Combined
MRR	1 char	0.154	0.140 (-9.3%)	0.190 (+22.9%)
	2 chars	0.256	0.326 (+27.2%)	0.375 (+46.4%)
	3 chars	0.380	0.469 (+23.5%)	0.530 (+39.6%)
	4 chars	0.465	0.527 (+13.3%)	0.582 (+25.2%)
	1 term	0.242	0.320 (+32.4%)	0.389 (+60.7%)
Success@5	1 char	0.199	0.206 (+3.8%)	0.266 (+33.9%)
	2 chars	0.317	0.450 (+41.9%)	0.494 (+55.6%)
	3 chars	0.470	0.588 (+25.1%)	0.653 (+38.9%)
	4 chars	0.552	0.638 (+15.4%)	0.694 (+25.7%)
	1 term	0.287	0.373 (+30.0%)	0.455 (+58.6%)

Table 2: Query-based MRR/Success@5 evaluation of our suggestion methods for different prefix lengths. Gains are compared to our baseline results listed in the *Logs* column.

	Prefix	Logs	Mailbox	Combined
NDCG	2 chars	0.20	0.41 (+99.7%)	0.41 (+98.9%)
	3 chars	0.31	0.62 (+97.2%)	0.64 (+106.2%)
	4 chars	0.47	0.71 (+49.4%)	0.76 (+60.2%)
Succ@5	2 chars	0.25	0.38 (+53.3%)	0.39 (+55.6%)
	3 chars	0.38	0.60 (+56.5%)	0.64 (+66.7%)
	4 chars	0.52	0.66 (+26.6%)	0.71 (+36.2%)

Table 3: NDCG/Success@5 manual evaluation results of our completion methods for different prefix lengths. Gains are compared to our baseline results listed in the *Logs* column.

to the user's needs. On the other hand, the query logs help bridge the query-document language gap, and, even if in a more moderate manner than in Web search, offer popular suggestions.

3.4 Manual Evaluation

In order to get some qualitative feedback, we asked ten Yahoo assessors to issue 20 queries each, using their own mailboxes, and rate the presented suggestions for a prefix of 2,3 and 4 characters. They judged whether each suggestion was well-formed or not, and whether it was relevant to their query intent (not relevant/relevant/highly relevant). We collected overall about 3,000 judgments of suggestions originating from one of three models (Logs, Mailbox and Combined). The results, as shown in Table 3, are mostly correlated with our other evaluations. In terms of usefulness, the assessors found our combined model to perform best, on average 88.4% better than Logs and 3.8% better than Mailbox. In about 40% of the queries, they found their intended query among the top 5 suggestions after typing just 2 characters, and in 70% of the cases after typing 4 characters. In addition, the assessors rated 93% of the suggestions from the combined model as well-formed.

3.5 Result-Based Evaluation

To complete the picture, we conducted a *result-based* evaluation of the log-based, mailbox-based and combined approaches. Given that there can be more than one relevant suggestion, we used the *MAP* metric for different prefix lengths, as presented in Table 4. This allows us to measure the overall search success, whatever query was used. We note that the combined suggester significantly outperforms both the log- and mailbox-based suggesters. The relative benefits of the mailbox as compared to users' logs are even higher here, increasing the chances to reach the desired message, and further confirming the success of our approach.

	Prefix	Logs	Mailbox	Combined
MAP	1 char	0.163	0.200 (+23.1%)	0.213 (+31.1%)
	2 chars	0.270	0.384 (+42.0%)	0.402 (+48.7%)
	3 chars	0.418	0.548 (+31.2%)	0.557 (+33.3%)
	4 chars	0.525	0.594 (+13.1%)	0.623 (+18.7%)
	1 term	0.252	0.347 (+38.0%)	0.388 (+54.1%)

Table 4: Results-based *MAP* evaluation of our suggestion methods for different prefix lengths. Gains are compared to our baseline results listed in the *Logs* column.

4 CONCLUSION

In this work, we considered the task of mail query completion, and presented a new approach for generating and ranking candidates, leveraging the mailbox content of the user. We demonstrated that due to the private and personal nature of the mail domain, the mailbox content of the individual user brings clear value. We used a range of mail-specific features at different levels of granularity (specifically, mailbox-, message- and field-based features), and analyzed their effectiveness. We conducted a manual evaluation and large scale experiments on real data from Yahoo Mail, and compared our mailbox-based suggestions to a recent log-based method proposed in [6]. In addition, we presented a combined approach that incorporates the benefits of both logs-based and mailbox-based methods, and significantly outperforms both.

REFERENCES

[1] Michael Bendersky, Xuanhui Wang, Donald Metzler, and Marc Najork. Learning from User Interactions in Personal Search via Attribute Parameterization. In *Proceedings of WSDM'2017*. Cambridge, United Kingdom.

[2] Adam Berger and John Lafferty. 1999. Information Retrieval As Statistical Translation. In *Proceedings of SIGIR'1999*. ACM, Berkeley, California, USA, 8.

[3] Sumit Bhatia, Debapriyo Majumdar, and Prasenjit Mitra. 2011. Query Suggestions in the Absence of Query Logs. In *Proceedings of SIGIR'2011*. ACM, Beijing, China.

[4] Fei Cai and Maarten de Rijke. 2016. A Survey of Query Auto Completion in Information Retrieval. *Foundations and Trends in IR* 10, 4 (2016), 273–363.

[5] David Carmel, Guy Halawi, Liane Lewin-Eytan, Yoelle Maarek, and Ariel Raviv. 2015. Rank by Time or by Relevance?: Revisiting Email Search. In *Proceedings of CIKM'2015*. ACM, Melbourne, Australia.

[6] David Carmel, Liane Lewin-Eytan, Alexander Libov, Yoelle Maarek, and Ariel Raviv. 2017. The Demographics of Mail Search and their Application to Query Suggestion. In *Proceedings of WWW'2017*. ACM, Perth, Australia.

[7] David Carmel, Liane Lewin-Eytan, Alexander Libov, Yoelle Maarek, and Ariel Raviv. 2017. Promoting Relevant Results in Time-Ranked Mail Search. In *Proceedings of WWW'2017*. ACM, Perth, Australia.

[8] Koby Crammer, Alex Kulesza, and Mark Dredze. 2013. Adaptive Regularization of Weight Vectors. *Machine Learning* 91, 2 (May 2013).

[9] Alan Feuer, Stefan Savev, and Javed A. Aslam. 2007. Evaluation of Phrasal Query Suggestions. In *Proceedings of the CIKM'2007*. ACM, Lisbon, Portugal.

[10] Mihajlo Grbovic, Guy Halawi, Zohar Karnin, and Yoelle Maarek. How many folders do you really need? Classifying email into a handful of categories. In *Proceedings of CIKM'2014*. ACM, Shanghai, China.

[11] David Hawking and Kathy Griffiths. An Enterprise Search Paradigm Based on Extended Query Auto-completion: Do We Still Need Search and Navigation?. In *Proceedings of ADCS '13*. Brisbane, Australia.

[12] Umut Ozertem, Olivier Chapelle, Pinar Donmez, and Emre Velipasaoglu. Learning to suggest: a machine learning framework for ranking query suggestions. In *Proceedings of SIGIR'2012*. ACM, Portland, Oregon, USA, 25–34.

[13] Pranav Ramarao, Suresh Iyengar, Pushkar Chitnis, Raghavendra Udupa, and Balasubramanyan Ashok. 2016. InLook: Revisiting Email Search Experience. In *Proceedings of SIGIR'2016*. ACM, Pisa, Italy.

[14] Milad Shokouhi. Learning to Personalize Query Auto-completion. In *Proceedings of SIGIR'2013*. ACM, Dublin, Ireland.