Query Expansion for Email Search

Saar Kuzi*
Technion, Israel
saarkuzi@campus.technion.ac.il

David Carmel
Yahoo Research, Israel
dcarmel@yahoo-inc.com

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

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

ABSTRACT

This work studies the effectiveness of query expansion for email search. Three state-of-the-art expansion methods are examined: 1) a global translation-based expansion model; 2) a personalized-based word embedding model; 3) the classical pseudo-relevance-feedback model. Experiments were conducted with two mail datasets extracted from a large query log of a Web mail service. Our results demonstrate the significant contribution of query expansion for measuring the similarity between the query and email messages. On the other hand, the contribution of expansion methods for a well trained learning-to-rank scoring function that exploits many relevance signals, was found to be modest.

CCS CONCEPTS

•Information systems → Information retrieval;

KEYWORDS

Email search, Query Expansion

1 INTRODUCTION

Searching over email data has attracted a lot of attention recently and several attempts have been made by the research community to apply up-to-date ranking paradigms for email search [2, 4]. In these paradigms, the relevance of the message to the query is estimated by a complicated scoring function that considers many signals of relevance, including the message freshness, the textual similarity to the query, the user interaction with the message, and many more signals [2]. However, email queries which are extremely short become a severe limitation for an accurate estimation of message relevance to the query. While the average query length on the Web is about three terms per query, the average length in the email domain is only 1.5 terms per query [2]. Query expansion techniques which expand the user's original query with related terms can presumably deal with the short query problem in email search.

Query expansion (QE) has long been recognized as an effective technique to improve retrieval performance, and has been studied for decades by the IR community for bridging the lexical gap between user queries and the searchable content [5, 9].

In this work we experiment with three complimentary query expansion approaches for email search. The first approach is based on learning a translation model from queries to messages [6]. Given a query log of a commercial Web mail service, we extract a large dataset of email queries, each associated with clicked (presumably relevant) messages. The data pairs are used to train a translation model that maps query terms to relevant message terms. Queries are then expanded by the most related terms to the query terms.

The translation model is based on the common query log of all users, hence a given query is expanded exactly the same for different searchers, ignoring user personal preferences and biases. However, email search is inherently personal; searching for a contact name for example has a totally different meaning for different searchers. Our second expansion model expands the query in a personal manner. The content of the personal mailbox of each user is used to train a word embedding model [11] which projects all terms of the messages into a dense lower dimensional space. The nearest neighbors of the query terms in the embedding domain are then used for expansion [8].

Both expansion models described so far are based on the global query log or on the user own mailbox. Both can be constructed in an offline manner. In contrast, relevance feedback based methods do not depend on any auxiliary data resource. Pseudo relevance feedback (PRF) methods construct an expansion model from the search results and then expand the query using the inferred model [9]. The expanded query can be used for re-ranking the search results, or to be submitted as a new search task [7]. In general, PRF methods improve the search effectiveness on average, however, they are very sensitive to the quality of the original search results.

Our work studies the contribution of the three different query expansion methods, described above, for email search effectiveness. We experimented with two large datasets of real user queries and corresponding messages, extracted from the query log of a Web mail system, and examined how each of the expansion methods affects the search performance. We show that while all expansion models bring benefit to the search effectiveness, the global translation based expansion model outperforms the two other methods. In the following we describe in details the three expansion models and our experimental results.

*This work was done during Saar Kuzi's internship at Yahoo Research.

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

DOI: http://dx.doi.org/10.1145/3077136.3080660

2 EXPANSION MODELS

2.1 Translation Model

Our first model expands the query using a global translation model, learned from the system's query log. Our query log items are composed from the user queries, each one is associated with a list of up to 100 retrieved messages, retrieved by the search system, where messages that were clicked by the user in the retrieved list are marked relevant. If there are more than one clicked message, we only consider the latest clicked one as relevant to the query.

Hence our training data includes $< Q, s >$ pairs extracted from the query log, where Q is the original user query, and s is the text of the subject of the clicked message for that query. We build a translation model from queries to subjects, using IBM model 1 [1], to be used for query expansion. The translation model provides, for each query term t, a probability distribution over the vocabulary V, i.e. $\forall w \in V, Pr(w|t)$ is the prior probability of "translating" term t to term w.

Given the learned translation model, we score the vocabulary terms for a given query Q_{orig} using the following formula:

$$Score(w; Q_{orig}) = \sum_{t \in Q_{orig}} \log(1 + Pr(w|t)). \quad (1)$$

We then select the top-k scored words for query expansion. The selected terms are added to the original query, weighted according to their translation score, where the term weight is normalized with respect to all k expanded terms, i.e., $s(w) = \frac{Score(w;Q_{orig})}{\sum_{i=1}^{k} Score(w_i;Q_{orig})}$.

Finally, the expanded query, $Q_{exp} = \{(w_1, s(w_1)), \ldots, (w_k, s(w_k))\}$, is linearly combined with the original query, where λ, the anchoring weight parameter, balances between the original query terms and the expanded terms,

$$Q_{final} = \lambda \cdot Q_{orig} + (1 - \lambda) \cdot Q_{exp}. \quad (2)$$

2.2 Personalized query expansion

The second expansion model we experimented with is based on the personal user mailbox. As users search over their own data, using their own personal vocabulary, expanded terms should reflect their own preferences [7].

In the personal query expansion model, the query is expanded with terms that are "semantically similar" to the query terms, where similarity is measured in the context of the personal user content. We used word embedding for measuring semantic similarity between terms. Specifically, we use the *Word2Vec* Continuous Bag-of-Words (CBOW) approach [11] which embeds terms in a vector space based on their co-occurrence in windows of text. The cosine similarity between the term vectors was shown to correlate with semantic similarity. Accordingly, we select terms similar to the query in this vector space for query expansion.

In our personal expansion model, the candidate terms are selected from the user own mailbox. The "Subject" field terms, and the "From" and "To" field terms of all user mailbox messages are represented by the *Word2Vec* model. The similarity of term w to query term t is determined by $Pr(w|t) = \frac{e^{\cos(\vec{t}, \vec{w})}}{\sum_{w'} e^{\cos(\vec{t}, \vec{w'})}}$, where $\cos(\cdot, \cdot)$ is the cosine between the two term vectors. The final term score for the given query is determined by aggregation over the query terms using Equation 1, and the top-k scored terms are then added to the query using Equation 2.

2.3 Pseudo Relevance Feedback

The final expansion model we experimented with is the classical pseudo relevance feedback (PRF) model [9]. In this model, the top scored messages retrieved for the query are used for constructing an expansion model.

in our study we focus on re-ranking the given search results. Given the list of retrieved messages, $M = (m_1, \ldots, m_n)$, the relevance model *RM1* [9] is used to construct the expanded query. *RM1* model is defined by $Pr(t|RM1) = \sum_{m \in M} Pr_{MLE}(t|m) Pr(m|q)$. $Pr(m|q)$ is estimated by the normalized message score for the query, while $Pr_{MLE}(t|m) = \frac{tf(t \in m)}{\sum_{t' \in m} tf(t' \in m)}$ is the maximum likelihood estimate of term t in message m; $tf(t \in m)$ is the number of occurrences of t in m. A message is defined here as a concatenation of its "Subject", "From", and "To" field terms.

Given the relevance model, we select the top-k scored terms according to the model and expand the query with them using Equation 2. The expanded query is then used to re-rank the search results.

2.4 Ranking Model

We would like to explore the potential merit of the query expansion approaches over an effective representation of the original query. To that end, we used two retrieval models. The first one measures only textual similarity of the query to the message based on the sequential dependence model (SDM) [10]. Each message in the collection is then scored with respect to the query using a linear interpolation of the *SDM* scores of its fields.

For the second retrieval model we used *REX*, a state-of-the-art learning-to-rank scoring function for email search [2], which considers the message freshness, the textual similarity to the query, and the user actions on the messages. The similarity of the expanded query to the message is added as an additional feature to the scoring function. During training, the LTR process learns an optimal weight for all components of the scoring function, including the expanded query.

3 EXPERIMENTS

3.1 Experimental Setup

Two query sets were used for experiments. Both of them contain queries collected from a Web mail system's query log. Each entry in the query log consists of a query text and the corresponding result list which was retrieved by the current search system and was exposed to the user during search time; we consider only search results that were received no more than six months before the query was issued, and only the latest clicked message in the result list is considered relevant.

In order to simulate the personal search conducted by email users we constructed a personal search index for each of the users in our training set. The searchable messages associated with a specific user were collected differently in the two query sets. In one set, denoted **LogData**, we collected the union of all messages of the user in the query log, i.e., all messages retrieved as a result

to at least one of the user's queries. In the second dataset, denoted **MailBox**, we used the mail data of a small group of users for which their mailboxes are fully available for us; all of the messages in the specific user's mail box constitute her personal collection.

For the **LogData** we randomly sampled 1659 users and collected their queries and related messages during a period of 20 days, resulting in 4446 queries each associated with at most 50 result messages. For the **MailBox** dataset we collected the personal messages of 100 users during a period of 34 days. The 2222 queries of these users, submitted during this period, and their corresponding feedback on the search results were collected from the query log. There is no overlap between users whose data is found in **LogData** and those in **MailBox**.

We used the open source Lucene package for searching over the personal collections (www.lucene.apache.org). Queries and messages were processed in the same manner. Stopwords were removed, and text was stemmed using Lucene's minimal English stemmer[1]. For the personalized expansion approach we trained a CBOW Word2Vec model for each of the personal collections[2]; free parameters were set to default values. For training the translation model we sampled queries of 11,200 users, over a period of 9 months, resulting in approximately 2.5M pairs of queries and subjects. These users are different from those whose data was collected for our query sets.

The performance of the different models is evaluated using two measures: Mean Reciprocal Rank (MRR) and *success@n*. $MRR = \sum_{Q \in S_Q} \frac{1}{r_Q}$; S_Q is the query set, and r_Q is the rank of the relevant message in the result list. *success@n* measures the percentage of queries in which the relevant message is among the top n messages; $n \in \{1, 5, 10\}$. We also report the reliability of improvement (RI), which measures the robustness of a query expansion approach with respect to using only the original query. $RI = \frac{|S_Q^+| - |S_Q^-|}{|S_Q|}$; S_Q^+ and S_Q^- are the sets of queries for which the *RR* of the expanded query is higher or lower, respectively, from the baseline of using the original query only. The two-tailed paired t-test at 95% confidence level is used in order to determine significant differences in performance.

The models that we study incorporate several free parameters. Free parameter values were set in the following manner. We split the query set at random to training (1/3) and test (2/3) sets. Then, we choose parameter values that maximize MRR on the training set, and report performance on the test set. The following values for free parameters were used. The number of terms used for expansion, k, is selected from $\{5, 10, 20\}$; λ, the anchoring parameter which balances between the original and the expanded query is in $\{0.0, 0.1, 0.5, 0.9, 1.0\}$; and the number of search results used for constructing RM1 for PRF is in $\{5, 10, 20\}$.

3.2 Experimental Results

3.2.1 Main result. The performance of the different expansion approaches is compared with those of the SDM model, on which they apply, in Table 1. We also study the Fusion approach which utilizes both the Personalized and the Translation models (see below for more details).

[1] org.apache.lucene.analysis.en.EnglishMinimalStemmer
[2] https://code.google.com/archive/p/word2vec

Dataset	Method	MRR	success@1	success@5	success@10	RI
MailBox	SDM	.264	.148	.384	.520	–
	PRF	.265	.152	.386	.513	**.213**
	Personalized	$.269^{i}$	$.155^{i}$	.388	$.528_{f}$	.116
	Translation	$.279^{i}_{f,p}$	$.162^{i}_{f}$	$.409^{i}_{f,p}$	$.540^{i}_{f}$	.138
	Fusion	$\mathbf{.281}^{i}_{f,p}$	$\mathbf{.163}^{i}_{f}$	$\mathbf{.412}^{i}_{p,f}$	$\mathbf{.548}^{i,g}_{p,f}$	.174
LogData	SDM	.278	.152	.404	.553	–
	PRF	$.284^{i}$	.156	.408	.560	**.228**
	Personalized	.281	.155	.402	.554	.135
	Translation	$.288^{i}_{p}$	$.158_{p}$	$\mathbf{.418}^{i}_{p}$	$.570^{i}_{p}$	.141
	Fusion	$\mathbf{.289}^{i,g}_{p}$	$\mathbf{.159}^{i}$	$\mathbf{.418}^{i}_{p}$	$\mathbf{.571}^{i}_{f,p}$	.149

Table 1: Applying query expansion on top of SDM. Significant differences with SDM, PRF, Personalized, and Translation are marked with 'i', 'f', 'p' and 'g', respectively.

According to the results, all query expansion approaches outperform SDM in a vast majority of reference comparisons; most of the MRR improvements are statistically significant. Among the three suggested approaches, Translation is the most effective according to Table 1. In terms of MRR it significantly improves over SDM by the largest margin in both datasets. Moreover, the performance of Translation (MRR and success@n) often dominates those of PRF and Personalized; most of the differences between them are statistically significant.

The Personalized expansion approach is effective in the case of MailBox. It outperforms SDM for all evaluation measures; the differences in MRR and success@1 are significant. Furthermore, it also outperforms PRF in most evaluation measures, albeit to a statistically significant degree only in one case. In LogData, on the other hand, the Personalized approach never outperforms SDM in significant manner. This difference in performance between the two datasets can be ascribed to the number of available messages per user which is larger in the case of MailBox. As the Word2Vec approach requires large amounts of data in order to learn an effective model, the resulted personalized models in the case of LogData may be less effective.

The PRF approach significantly improves SDM in terms of MRR only in LogData. The difference in performance of the two data sets can be attributed to their different nature. The collection of messages per user in the case of LogData is built from the search results (obtained by the current system). Hence, it is likely that the initial result list, from which RM1 is induced, is of better quality in LogData as compared to the search results obtained from MailBox, which were extracted from all user messages. In addition, PRF is the most robust expansion approach in terms of RI. A possible explanation for this is that PRF models are induced from an initial result list. Such models, therefore, tend to be closer to the original query as compared to other approaches that utilize some external source (which is constructed in a non-query dependent manner). Consequently, PRF models may be less risky.

Fusion. As already noted, the Personalized and Translation approaches are of complementary nature. Empirically, we found that an "oracle" which chooses between the two expansion approaches on a per query basis (using RR), yields over 10% MRR improvement in both data sets. Motivated by this finding, we suggest the Fusion approach. In Fusion, the term lists of both approaches are linearly interpolated using a free parameter learned from $\{0.0, 0.1, 0.5, 0.9, 1.0\}$. Specifically, we fuse expansion term

Dataset	Method	Person	Company	Content	All
MailBox	SDM	.262	.176	.292	.264
	PRF	.264	.182	.291	.265
	Personalized	$.272^{i}$	.172	.294	$.269^{i}$
	Translation	$\mathbf{.274}^{i}$	$\mathbf{.206}^{i}_{p}$	$.306^{i}_{f,p}$	$.279^{i}_{f,p}$
	Fusion	$\mathbf{.274}^{i}$	$.201^{i}_{p}$	$\mathbf{.307}^{i}_{f,p}$	$\mathbf{.281}^{i}_{f,p}$
LogData	SDM	.290	.214	.286	.278
	PRF	**.296**	.209	$.294^{t}$	$.284^{t}$
	Personalized	.292	.219	.288	.281
	Translation	$.288^{i}_{p}$	$\mathbf{.254}^{i}_{f,p}$	$.298^{i}_{p}$	.288
	Fusion	.289	$\mathbf{.254}^{i}_{f,p}$	$\mathbf{.299}^{i}_{p}$	$\mathbf{.289}^{i,g}_{p}$

Table 2: MRR of three query groups: People, Company, and Content. Significant differences with SDM, PRF, Personalized, and Translation are marked with 'i', 'f', 'p' and 'g', respectively.

Dataset	Method	MRR	success@1	success@5	success@10	RI
MailBox	Time	.364	.220	.538	.668	–
	REX	.423	.280	.586	.718	–
	PRF	$.433^{t}$	.286	$.612^{t}$	.724	.287
	Personalized	.431	.278	$.615^{i}$	.729	.279
	Translation	$\mathbf{.446}^{i}_{f,p}$	$\mathbf{.299}^{i}_{f,p}$	$\mathbf{.620}^{i}$	$\mathbf{.735}^{i}_{f}$	**.298**
	Fusion	$.440^{i}_{f,p}$	$.293_{p}$	$.618^{i}$	$.726^{g}$	.290
LogData	Time	.387	.245	.552	.669	–
	REX	.436	.293	.611	.725	–
	PRF	.435	.294	.609	.726	**.170**
	Personalized	.437	.292	$.615^{i}$	.726	.073
	Translation	$\mathbf{.441}^{i}$	**.296**	$\mathbf{.620}^{i}_{f}$	$\mathbf{.732}^{i}$	.120
	Fusion	$.438^{i}$	.294	$.613^{g}$	$.724^{g}$	.071

Table 3: Applying query expansion on top of REX. Significant differences with REX, PRF, Personalized, and Translation are marked with 'i', 'f', 'p' and 'g', respectively.

lists of 50 terms of Translation and Personalized, and then extract k terms with the highest combined score for expansion (using Equation 2). We can see in Table 1 that the performance of Fusion dominates those of the two approaches. However, the difference with Translation is significant only in one case. In terms of RI, we can see that Fusion is much more robust than both approaches. Given the high performance of the oracle, the question of how to optimally combine the two approaches remains open.

Further analysis. Queries in email search can be categorized according to their intent. In this work, we divide the queries into three intents: Person (looking for messages of a specific contact), Company (messages of a specific company), and Content (all the rest). The query intent was detected by an existing intent classifier [3]. Person, Company, and Content queries are (approximately) 40%, 10% and 50%, of all queries in both data sets, respectively. In order to further explore the effectiveness of query expansion approaches, we examined their MRR performance separately, in each category. Table 2 summarizes our main findings.

In both datasets the best performance of Translation is attained for company queries; the improvements over SDM are significant and by more than 17%. A possible explanation is that Translation is learned using query logs of many users. Company queries and the corresponding relevant messages are often common among different users since these messages are usually generated by machine. Therefore, the resultant translation model is better in generalizing over users.

As for PRF and Personalized, the findings are not consistent over the datasets. Personalized is more effective in MailBox and it significantly improves SDM by more than 3%. As Person queries often reflect different information needs of users, the personalized model is effective for such queries. PRF, on the other hand, is more effective in LogData and its effectiveness is attained both for Person and Content queries.

3.2.2 Integration with REX. The performance of the different approaches when applied on REX is presented in Table 3. We also present the performance of Time, an approach in which messages are ordered according to their received date. Such approach is very common in commercial email services.

As in the case of SDM, Translation is the most effective expansion approach over REX in both datasets. In MailBox we can see that all methods outperform REX in most evaluation measures; most MRR and all success@5 improvements are significant. Moreover, the methods' effectiveness is robust in terms of RI. In LogData, on the other hand, while all methods, except PRF, improve over REX in terms of MRR, the improvements are significant only in two cases. A possible reason for this modest improvement is that in LogData the original ranking of REX is presumably of higher quality. In such a case, the added value of expansion terms to the high quality ranking function might be limited.

4 CONCLUSIONS

This on-going work studies the effectiveness of query expansion methods for email search. All the expansion methods outperform a standard SDM based ranking model in significant manner, and the translation model outperforms the other expansion methods. However, the amount of improvement over REX, a mature and well trained LTR ranking model, was found to be only modest. The questions how to optimally integrate the expanded query into learning-to-rank model and how to optimally fuse the expansion models are far from being solved and are left for future work.

REFERENCES

[1] Adam Berger and John Lafferty. 1999. Information Retrieval As Statistical Translation. In *Proceedings of SIGIR*. 222–229.

[2] 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*. ACM, 283–292.

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

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

[5] Claudio Carpineto and Giovanni Romano. 2012. A Survey of Automatic Query Expansion in Information Retrieval. *ACM Comput. Surv.* 44, 1 (Jan. 2012), 1:1–1:50.

[6] Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. 2002. Probabilistic Query Expansion Using Query Logs. In *Proceedings of WWW*. ACM, 325–332.

[7] Nick Craswell Fernando Diaz, Bhaskar Mitra. 2016. Query Expansion with Locally-Trained Word Embeddings. In *Proceedings of ACL*.

[8] Saar Kuzi, Anna Shtok, and Oren Kurland. 2016. Query Expansion Using Word Embeddings. In *Proceedings of CIKM*. ACM, 1929–1932.

[9] Victor Lavrenko and W. Bruce Croft. 2001. Relevance Based Language Models. In *Proceedings of SIGIR*. ACM, 120–127.

[10] Donald Metzler and W Bruce Croft. 2005. A Markov random field model for term dependencies. In *Proceedings of SIGIR*. ACM, 472–479.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv:1301.3781* (2013).