Multi-Word Generative Query Recommendation Using Topic Modeling

Matthew Mitsui
Department of Computer Science
Rutgers University
New Brunswick, NJ, USA, 08901
mmitsui@rutgers.edu

Chirag Shah
School of Communication and Information
Rutgers University
New Brunswick, NJ, USA, 08901
chirags@rutgers.edu

ABSTRACT

Query recommendation predominantly relies on search logs to use existing queries for recommendation, typically calculating query similarity metrics or transition probabilities from the log. While effective, such recommendations are limited to the queries, words, and phrases in the log. They hence do not recommend potentially useful, entirely novel queries. Recent query recommendation methods have proposed generating queries on a topical or thematic level, though current approaches are limited to generating single words. We propose a hybrid method for constructing multi-word queries in this generative sense. It uses Latent Dirichlet Allocation to generate a topic for exploration and skip-gram modeling to generate queries from the topic. According to additional evaluation metrics we present, our model improves diversity and has some room for improving relevance, yet offers an interesting avenue for query recommendation.

CCS Concepts

•**Information systems** → **Query suggestion;** *Personalization;* Document topic models; Retrieval effectiveness;

Keywords

Search session analysis; Diversity; User simulations; Exploratory search; Query recommendation; Latent Dirichlet Allocation

1. INTRODUCTION

Query recommendation dates back to least the early 2000s. Most approaches harness search logs - a set of users' search sessions - to make personalized recommendations to users. Each session contains information about a user's sequence of queries, clicks on search engine result pages (SERPs), and possibly time spent on results. Even without explicit relevance feedback on the logs from annotators, this data provides a rich source of implicit feedback - such as click-through behavior and dwell time - that can be used to quantify query satisfaction, page usefulness and pairwise query similarity scores. While query logs are indispensable for personalization, many algorithms either recommend queries directly from the logs or reformulated versions of prior queries. While effective on a large query log, this may work less well for smaller data. For recommending rarely-discovered, topic-relevant information or making session-based recommendations, long-tail queries become less likely to produce satisfactory results, necessitating novel query generation. This contrasts with reformulation approaches that suggest textually similar queries, recommending queries purely on topical relatedness to previous queries in a session.

In this paper, we discuss an approach to recommending novel queries that combines a user's session and the thematic structure of web pages to create new queries. We create recommendations using Latent Dirichlet Allocation and skip-gram modeling and apply new evaluation metrics to this problem setting. In Section 2, we discuss the necessary background information to understanding this work. In Section 3, we explain our algorithm. We outline our data set in Section 4. We give our performance metrics, results, and discussion in Section 5, and we conclude in Section 6. While there is room for improvement, our model improves diversity of actual performance and offers interesting possibilities.

RecSys '16, September 15-19, 2016, Boston , MA, USA

DOI: http://dx.doi.org/10.1145/2959100.2959154

2. BACKGROUND

2.1 Query Recommendation

Some past and current query recommendation methods recommend queries directly from a query log. [1], for instance, clusters queries by similarity, offering similar substitutes for failing queries, and [4] offers suggestions by calculating explicit transition probabilities between queries in a log. Contrastingly, [11] uses query distance to recommend orthogonal queries - dissimilar queries that are similar enough to be topically relevant. Some methods generate new queries incrementally. [5], for instance, uses WordNet to replace words with their synonyms. While our approach recommends orthogonal queries like [11], unlike [11] and [5] it generates queries without incrementing from a base string.

Hence, our approach risks retrieving less relevant results. Non-zero transition probabilities in a query log give implicit relevance feedback. Our approach discards this feature for relevance in our algorithm (assuming no multitasking).

2.2 Topic Modeling in Query Suggestion

We attempt to address this loss of relevance feedback and capture diversity with topic models, modeling the topics a

user has explored and recommending terms from unexplored topics. For at least the past decade, topic modeling has gained recent interest among researchers as an alternative approach to modeling documents. A commonly cited topic model is Latent Dirichlet Allocation (LDA) [2], which can infer the topic distribution of a document and bag-of-words distribution of a topic. [6, 8] have previously applied LDA to recommending queries. [6] only mapped existing queries to a thematic representation and recommend those queries. [8] only recommends single-word queries and generates virtual training "documents" by combining query sessions with an identical ending. This cannot be done with smaller data and may be insufficient for *exploratory search tasks* (see Task and Dataset). Our approach extends these efforts.

2.3 Evaluation Metrics

Typically, results of query recommendation are evaluated in a URL-based fashion. For instance, discounted cumulative gain sums up the relevance scores of each document in a list (e.g. query results) and tempers them according to their position in the list. Recent efforts like [9] evaluate whole session metrics; in particular, [9] defined URL-based measures like the coverage, unique relevant coverage, and likelihood of discovery of URLs for individual and collaborative tasks. Since our approach recommends new queries, it is likely that on smaller data, most URLs in a SERP have never been seen in our sessions. To our knowledge, most of the literature performs purely URL-based evaluation, so we define different effectiveness measures in our results.

3. ALGORITHM

We first train our LDA topic model $T = (\alpha, \beta_{1:K})$ over D, a set of web documents. α is a Dirichlet prior, and $\beta_k \in \beta_{1:K}$ a "topic", or a distribution over the vocabulary. For this paper, we trained on all of the pages linked through the top 10 URLs of SERPs in our log. All text - SERP snippets and web page text - is preprocessed through stopword removal and removing words with less than 4 characters. We also remove punctuation, numbers, and words with non-alphanumeric letters (e.g., hashtags).

Assume a user u in our query log has issued queries $Q_u = \{q_1, q_2, ..., q_h\}$ and has visited pages P_u in their entire session. We create simulations starting at each step $3 < t < h$; the completion of a simulation results in several injections $\{q_1, q_2, ..., q_t, q'_{t+1}, q'_{t+2}, ...\}$. At each step of the simulation, we concatenate the current SERP text (i.e. result snippets) of $\{q_1, q_2, ..., q_t\}$ and infer a topical distribution using T - call this distribution $\theta_{S,1:S,K}$. This tells us the proportions in which each topic has currently been explored by the user. We then select a topic for query generation and then generate the query (details discussed in Sections 3.1-3.2).

We issue the generated query q'_{t+1} using a commercial search API and inject it into the user's current set of queries, making $\{q_1, q_2, ..., q_t, q'_{t+1}\}$. We repeat the topic inference and injection loop until a timer in our algorithm reaches $|Q_u| + |P_u|$, the total number of pages visited and queries issued by user u in our actual log data. This stopping criterion simplifies and accommodates for individual user factors, such as page viewing time and user frustration. Each simulated query counts as a timestep, as do the first $|S|$ results, where $|S|$ is the size of a single SERP. See Algorithm 1 for a summarization of the algorithm.

3.1 Mining the Next Topic

After calculating $\theta_{S,1:S,K}$, we iterate through topics in ascending probability, to search for underrepresented yet relevant topics. We calculate topic k's relevance with a weighted divergence score against the SERPs' topic distribution:

$$Score(k) = \sum_{i=1}^{K} \theta_{S,i} \times JSD(\beta_i \parallel \beta_k) \quad (1)$$

Where JSD is the Jensen-Shannon Divergence score [7], a symmetric score based on KL-divergence. The first topic to score below 0.15 (chosen through trial and error) is chosen for query generation. We incrementally increase the threshold each time no satisfactory topics are found.

3.2 Query Generation from a Topic

After topic selection, we extract the topic's top 15 words and their topic-specific probabilities, as given by LDA. We generate 3-word queries randomly, sampling the first word from the topic probabilities. The remaining 2 words are then sampled from the skip-gram probabilities, conditioned on the first word. Skip-grams generalize n-grams; words are allowed to occur within a window of k words in order to be counted in the model. For smaller data, skip-grams can cope with data sparsity that can affect n-gram models, and they can be an effective smoothing method for language model estimation [10]. We used a 4-skip-3-gram model, i.e., we calculated the probabilities of sequences of 3 words (the most common query length in our data) that were no more than 4 words apart from each other. We trained our model (SG) on the titles of the web pages that were used for LDA, rather than the whole content; many non-stop words are irrelevant while titles tend to be related to the query.

Data:
T = (α,$\beta_{1:K}$), SG, Q_u, P_u, $|S|$- See Section 3 for explanations
ts - current timestamp
Q_{curr} - queries currently issued by a user
$index_{best}$ = Null;
while *ts* $< |P_u| + |Q_u|$ **do**
  $\theta_{S,1:S,K}$ = T.evaluate(ConcatenateSERPs(Q_{curr}));
  for $\theta_{S,k} \in \{min(\theta_{S,1:S,K}), ..., max(\theta_{S,1:S,K})\}$ **do**
    if *getWeightedDivergence*($\theta_{S,k}$,$\beta_{1:K}$) $< .15$ **then**
      $index_{best} = index(\theta_{S,k})$;
      break;
    end
  end
  $qterms$ = BestTopicWords($\beta_{index_{best}}$,SG);
  q_{new} = IssueQuery($qterms$);
  $Q_{curr} = Q_{curr} + \{q_{new}\}$;
  ts += $|S| + 1$;
end
return Q_{curr}

Algorithm 1: Iterative version of the algorithm.

4. TASK AND DATASET

We applied our approach to *exploratory search task* data. Exploratory search tasks have been estimated to comprise 10% of search sessions and 25% of overall queries [3]. In these tasks, information needs are ill-structured and cannot

Topic	# users	> 3 SERPs	$\|Q_{uni}\|$	Avg. queries	$\|D\|$
TEC	13	9	99	12.63	513
HEA	16	15	148	10.13	915
ENT	11	11	159	14.72	581
ART	5	5	91	16	740
ENV	2	2	28	14	115

Table 1: Users, queries, and documents per topic

Algorithm	TEC	HEA	ENT	ART
Session	3.08	0.57	0.41	0.84
ORTH	3.01	2.06	0.69	1.81
LDA-SG	3.44	1.06	0.44	0.78

Table 2: Percent of on-topic words

be satisfied in a single query, and users engage in learning during the task [13]. Our queries could hit poorly explored areas in such lengthy tasks, increasing diversity while maintaining relevance. In our data, 29 users completed 47 exploratory search sessions on 5 topics: "Technology" (TEC), "Health and Wellness" (HEA), "Entertainment" (ENT), "Art and History" (ART), and "Environment and Energy" (ENV). Users were undergraduate participants recruited through open calls in e-mail lists at a local university, from majors such as Computer Science, Information Technology, Informatics, Nursing and Biological studies. Users completed a prompt within a 2-hour time limit and were provided monetary compensation. They were allowed to quit after a minimum of 1 hour but were incentivized with an additional reward for top performance. Users downloaded a Firefox plugin and conducted the task live through their web browsers, i.e., at any location at their convenience. Despite lack of lab supervision, users can conduct more natural searching behavior. Users selected search topics from a brief list ("Technology", "Health and Wellnes", etc.). After selection, they saw a full task description and could not retract their choices. They had a 5-minute warm up task before choosing the main task.

Our plugin recorded basic browsing behavior, like changing tabs and issuing queries. It had buttons for snipping page text that users highlight, reviewing the task description, and composing a final report on an online text editor. The plugin collected the content of users' SERP and web page visits live, by issuing curl and search API calls from our server, approximating results users saw at visitation time. We created topics we thought were unfamiliar enough to be unaffected by personalization from users' browsers.

Here is the task description (paraphrased) for TEC: "With the recent focus on security breaches and consumer data breaches, vulnerability in data and software has become an important topic of interest. You have been asked to write an article about data and software vulnerabilities; it should be around 1000 words. It should appeal to as many people as possible, including an unfamiliar, lay audience and people in affected businesses. It should cover different aspects of software vulnerabilities that are relevant to their daily lives. It should also focus on the measures taken to minimize these risks at industry, governments and consumer levels. You can use snippets to collect information that you deem useful for writing the report and copy them into your article."

Table 1 shows user and query counts for each topic. We trained a single LDA model once over the entire corpus of

Algorithm	TEC	HEA	ENT	ART
Session	11.63	14.74	7.89	7.35
ORTH	9.97	26.34	16.42	40.68
LDA-SG	9.21	20.03	8.39	31.02

Table 3: Percent of relevant ODP topics

Unit of Measure	TEC	HEA	ENT	ART
Words	0.06	1.11	1.15	0.83
ODP topics	11.01	15.73	12.69	4.32

Table 4: Percent of off-topic words and ODP topics covered by LDA-SG. (Results for LDA-SG only.)

documents - the documents linked through users' first pages in their SERPs. We did this to simulate a real web environment absent of supervised topic labels on pages may not be available. We set the number of topics $K = 35$. We only used sessions with over 3 queries; others do not provide sufficient search results for our algorithm. We then filtered topics with less than 3 users as well. Many users issued over 3 queries, providing motivation for this algorithm.

5. RESULTS AND ANALYSIS

We compared our algorithm (LDA-SG) against the real user sessions ("Session" in the tables) and Vahabi et al. [11] (ORTH). ORTH is strictly recommends queries from the logs to diversify results. ORTH recommends queries that overlap with the most recent query by (0.0,0.06]. Due to sparse logs we incremented the ceiling of 0.06, eventually choosing a random candidate if no suitable ones could be found.

In light of Section 2.3, we chose lexical and ontological evaluation metrics instead of URL-based ones. Suppose we let $Words(Queries)$ be the set words in a sequence of queries (including SERP snippets), $E\_Words(T_i)$ be the set of topic-exclusive words for topic T_i, and $W = E\_Words(T_i) - \bigcup_{j \neq i} E\_Words(T_j)$ our score Cov(Queries,i) is $|Words(Queries) \cap W|$ divided by $|W|$. This measures the topic word coverage, or the percentage of words that a set of SERPs covers in topic i. If we substitute $W = \bigcup_{j \neq i} Words_{T_j}$ in Equation 2, we get our second measure: the percentage of off-topic words that are covered. The former measures diversity. The latter measures relevance as a penalty score, since the word "diabetes", for instance, belongs exclusively to the "Health and Wellness" topic and none of the others. We omit off-topic scores for actual user sessions; by definition, these results are all 0%. We similarly omit ORTH results for because all off-topic queries have 0.0 overlap due to a small query log.

Our ontological approach uses the Open Directory Project (ODP)[1] to convert URLs to topics. ODP is a web ontology that categorizes websites into hand-tailored topics, such as "Computers/Internet/Resources/Research" and has been used to categorize URLs for evaluation [14]. We convert all URLs in our simulations to their ODP categories (truncating URLs incrementally until a matching category can be found). Replacing words in the above equations with ODP categories gives us our ODP-based measures.

Our results are given in Tables 2-6. Our approach increases the percentage of topic-specific words and ODP categories (with a couple exceptions) over the baseline, suggest-

[1]http://www.dmoz.org

Table 5: Query sequences returned by our algorithm. Boldfaced queries are generated queries. The first sequence shows a shift from the original topic.

diet for type 2 diabetics, type 2 diabetes, risks of diabetic diet, exercise plans for type 2 diabetes, **diet healthy-eating clinic**, **information control solutions**, **animation degree campus-based**, **health guidelines micronutrients**, **diabetes stick-to-it plan**
diets for diabetics type 2, type 2 diabetes, type 2 diabetes causes, **diet claims clinic**, **health aetna featuring**, **diabetes weight loss**, **people diabetic cookbook**

Table 6: Top 10 words of returned LDA topics. Starred words belong to a sub-corpus (e.g. health) less dominant in the list (e.g. technology)

Topic 1: 'animated', 'animation', 'avatar', 'characters', 'computer', 'data'*, 'diabetes'*, 'disney', 'film', 'information'*
Topic 8: 'blood', 'body', 'data'*, 'diabetes', 'diet', 'exercise', 'film'*, 'food', 'glucose', 'health'

ing an increase in diversity. However, the our approach is not as diverse ORTH and is penalized on off-topic measures.

Interestingly, the penalty for off-topic words is small while the penalty for off-topic ODP categories is large. Off-topic words are those that exist *exclusively* in other sub-corpora, and likewise for off-topic ODP categories. Our scores suggest that many of the queries (and hence web pages) were from other topics, even though few of the words were exclusively from other topics. Assuming that topics do not mix in a single set of SERP results, this suggests that there is a large percentage of overlapping words between topics. This would suggest a shift between topics in our list of suggested queries, which is consistent with our data. See Tables 5 and 6. This suggests the need for perhaps a stronger method of topic segmentation that discounts for overlapping words.

6. CONCLUSION AND FUTURE WORK

In future work, we would need to address arbitrary parameter values chosen for proof of concept. Though obvious extrinsic evaluation metrics for cross-validation, like URL-based nDCG, are eliminated. Retrievability scores could fit but require large search logs (or simulations) for score estimates. We would have inferred the optimal number of topics with the online Hierarchical Dirichlet Process [12], but to our knowledge, implementations take the number of topics as a "truncation level" parameter. Also, the number of generated query terms could be probabilistically modeled from the log.

We could also improve our skip-gram method to generate queries more closely resembling human ones. We neglect the linguistic and information roles of stop words in real human queries (e.g. "rainbow" vs. "over the rainbow"), which would require further analysis. We could then add human judgments for evaluation, such as the comprehensibility of queries and whether they seemed to be human-generated.

We extended topic modeling-based methods of novel query generation. Our multi-word approach uses web documents as a lexical resource and attempts to maintain relevance that can be lost with topically-based generative approaches. We achieved higher diversity at the cost of relevance, and one desideratum of recommendation research is to close this gap while maximizing both. We believe this approach offers an interesting direction for query recommendation. If more widespread, topical recommendations could not only make recommendations at a topical level but could also increase the pool of unique queries in a log if users were to take the recommendations, improving non-generative methods.

7. ACKNOWLEDGMENTS

A portion of this work was supported by the US Institute of Museum and Library Services (IMLS) Career Development grant # RE-04-12-0105-12.

8. REFERENCES

[1] R. Baeza-Yates, C. Hurtado, and M. Mendoza. Query recommendation using query logs in search engines. In *Proceedings of the 2004 International Conference on Current Trends in Database Technology*, EDBT'04, pages 588–596. Springer-Verlag, 2004.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[3] D. Donato, F. Bonchi, T. Chi, and Y. Maarek. Do you want to take notes?: Identifying research missions in yahoo! search pad. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 321–330, New York, NY, USA, 2010. ACM.

[4] Q. He, D. Jiang, Z. Liao, S. C. H. Hoi, K. Chang, E.-P. Lim, and H. Li. Web query recommendation via sequential query prediction. In *Proceedings of the 2009 IEEE International Conference on Data Engineering*, ICDE '09, pages 1443–1454, Washington, DC, USA, 2009. IEEE Computer Society.

[5] U. Hermjakob, E. Hovy, and C.-Y. Lin. Automated question answering in webclopedia: a demonstration. In *Proceedings of the second international conference on Human Language Technology Research*, pages 370–371. Morgan Kaufmann Publishers Inc., 2002.

[6] L. Li, G. Xu, Z. Yang, P. Dolog, Y. Zhang, and M. Kitsuregawa. An efficient approach to suggesting topically related web queries using hidden topic model. *World Wide Web*, 16(3):273–297, 2013.

[7] J. Lin. Divergence measures based on the shannon entropy. *Information Theory, IEEE Transactions on*, 37(1), 1991.

[8] S. Momtazi and F. Lindenberg. Generating query suggestions by exploiting latent semantics in query logs. *Journal of Information Science*, pages 1–12, 2015.

[9] C. Shah. Evaluating collaborative information seeking - synthesis, suggestions, and structure. *J. Inf. Sci.*, 40(4):460–475, Aug. 2014.

[10] N. Shazeer, J. Pelemans, and C. Chelba. Skip-gram language modeling using sparse non-negative matrix probability estimation. *CoRR*, abs/1412.1454, 2014.

[11] H. Vahabi, M. Ackerman, D. Loker, R. Baeza-Yates, and A. Lopez-Ortiz. Orthogonal query recommendation. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 33–40. ACM, 2013.

[12] C. Wang, J. W. Paisley, and D. M. Blei. Online variational inference for the hierarchical dirichlet process. In *International conference on artificial intelligence and statistics*, pages 752–760, 2011.

[13] B. M. Wildemuth and L. Freund. Assigning search tasks designed to elicit exploratory search behaviors. In *Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval*, HCIR '12, pages 4:1–4:10, New York, NY, USA, 2012. ACM.

[14] X. Zhu, J. Guo, X. Cheng, P. Du, and H.-W. Shen. A unified framework for recommending diverse and relevant queries. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 37–46, New York, NY, USA, 2011. ACM.