Temporal Query Expansion Using a Continuous Hidden Markov Model

Jinfeng Rao
Department of Computer Science
University of Maryland, College Park
jinfeng@cs.umd.edu

Jimmy Lin
Cheriton School of Computer Science
University of Waterloo
jimmylin@uwaterloo.ca

ABSTRACT

In standard formulations of pseudo-relevance feedback, document timestamps do not play a role in identifying expansion terms. Yet we know that when searching social media posts such as tweets, relevant documents are bursty and usually occur in temporal clusters. The main insight of our work is that term expansions should be biased to draw from documents that occur in bursty temporal clusters. This is formally captured by a continuous hidden Markov model (cHMM), for which we derive an EM algorithm for parameter estimation. Given a query, we estimate the parameters for a cHMM that best explains the observed distribution of an initial set of retrieved documents, and then use Viterbi decoding to compute the most likely state sequence. In identifying expansion terms, we only select documents from bursty states. Experiments on test collections from the TREC 2011 and 2012 Microblog tracks show that our approach is significantly more effective than the popular RM3 pseudo-relevance feedback model.

1. INTRODUCTION

A longstanding challenge in information retrieval is the issue of vocabulary mismatch, where query terms are not present in relevant documents. This problem is especially severe in searching social media posts such as tweets due to their short lengths and frequent use of informal language. Query expansion techniques, especially those based on pseudo-relevance feedback, are effective in addressing this problem. The main idea is to augment the user's query with terms that appear in the initial top k retrieved documents. In this paper, we extend this idea to consider the temporal dimension in the term expansion process.

We are motivated by Efron et al.'s temporal cluster hypothesis [7], which stipulates that in search tasks where time plays an important role (such as tweet search), relevant documents tend to cluster together in time, and that this property can be exploited to improve search effectiveness. Just as van Rijsbergen's "classic" cluster hypothesis suggests that documents relevant to a query form clusters in term space, Efron et al. suggest that documents relevant to a query will form clusters along a timeline.

The temporal cluster hypothesis is illustrated by the visualizations in Figure 1, similar to those presented by Efron et al. [7], which help illustrate the intuition behind our techniques. These visualizations show three topics from the TREC 2011 Microblog track. In each timeline, the query time (the time at which the query was issued) is anchored to the right edge; the x-axis shows time prior to the query time, in days. Dots show tweets that were retrieved by participating teams and evaluated by assessors (i.e., the pools): green dots are relevant, red dots are highly relevant, and gray dots are not relevant. The underlying blue bars show the distribution of relevant and highly-relevant tweets as a histogram. As we can see, relevant tweets for topics 14 and 30 tend to cluster together in time, while relevant tweets for topic 6 are more evenly distributed. Across all topics from the TREC test collections, we observe many timelines that exhibit temporal clustering (like topics 14 and 30).

In standard formulations of pseudo-relevance feedback, the timestamp of a document is not considered in identifying expansion terms—yet we know from Figure 1 that relevant documents are bursty and usually occur in temporal clusters, and that this signal should be incorporated into the relevance feedback model. The main insight of our work is that term expansions should be biased to draw from documents that occur in the bursty temporal clusters. This is formally captured by a continuous hidden Markov model (cHMM), in which the temporal distribution of documents (not necessarily relevant) is represented by a sequence of hidden states; the probability of generating a particular number of documents from each state follows a Gaussian distribution. We present the derivation of an EM algorithm to estimate the parameters of such a cHMM. Given a query, we first perform an initial retrieval, estimate the parameters for a cHMM that best explains the observed distribution of retrieved documents, and then use Viterbi decoding to compute the most likely state sequence. In identifying term expansions, we only select documents from bursty states. Experimental evaluations on test collections from the TREC 2011 and 2012 Microblog tracks show that our approach is significantly more effective than the popular RM3 pseudo-relevance feedback model [10, 1].

2. RELATED WORK

There is a long thread of research exploring the role of temporal signals in search [11, 6, 5, 7, 9], and it is well estab-

ICTIR '16, September 12 - 16, 2016, Newark, DE, USA

ISBN 978-1-4503-4497-5/16/09...$15.00

DOI: http://dx.doi.org/10.1145/2970398.2970424

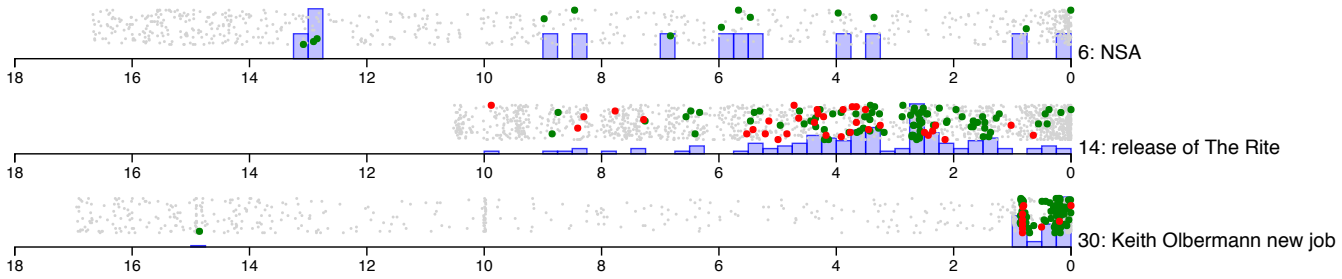

Figure 1: Temporal distribution of relevant (green) and highly-relevant (red) tweets for three topics from the TREC 2011 Microblog track.

lished that for certain tasks, better modeling of the temporal characteristics of queries and documents can lead to higher retrieval effectiveness. For example, Li and Croft [11] introduced recency priors that favor more-recent documents; Efron et al. [6] presented several language model variants that incorporate temporal evidence; Dakka et al. [5] proposed a moving window-based approach to integrate query-specific temporal evidence with lexical evidence.

More recently, Efron et al. [7] introduced the temporal cluster hypothesis (as discussed above), which was operationalized in retrieval models based on kernel density estimation. This work was subsequently expanded upon by Rao et al. [13]. We further build on this work using cHMMs to capture temporal evidence in the form of bursty documents. As such, our technique is most directly related to the work of Kleinberg [9], who proposed a weighted-automaton model to uncover bursty and hierarchical structure in document streams of email and news articles.

There have been several other works that studied temporal query expansion [8, 4, 3, 12]. Keikha et al. [8] represented queries and documents with their normalized term frequencies in the time dimension and used a time-based similarity metric to measure relevance. Craveiro et al. [4] exploited the temporal relationship between words for query expansion. Choi et al. [3] presented a method to select time periods for expansion based on users' behaviors (i.e., retweets). Our work is related to that of Peetz et al. [12], who identified temporal bursts through heuristics by setting hard thresholds on the distribution of document counts within a time window. Their approach showed improvements over a query-likelihood baseline and an exponential-decay baseline. We believe our cHMM approach represents a more formal and flexible way to capture bursty document behavior.

3. APPROACH

Previous work has established the importance of temporal signals in searching social media posts, and in this paper we apply this basic idea to pseudo-relevance feedback. Our main insight is that term expansions should be biased to draw from documents that occur in bursty temporal clusters of documents. We use a continuous hidden Markov model to identify such temporal clusters based on top-ranked documents from the user's initial query, and then compute the most likely underlying state sequence. Documents in the bursty state are then selected for query expansion. In this section, we present our formal model, describe an EM algorithm for estimating parameters, and explain how term expansions are computed.

3.1 Continuous Hidden Markov Model

We begin with the standard definition of an HMM for modeling a discrete observation sequence O of length T with a fixed number of hidden states. An HMM is parameterized by (A, B, π), where A is the transition matrix with A_{ij} denoting the transition probability from state i to state j at each time step, B is the emission matrix with each $B_i(O)$ denoting the probability of generating observation O from state i, and π is the initial state distribution vector.

Our approach is a variant of classic HMMs. In classic HMMs each observation is a discrete symbol drawn from a finite alphabet, while in our case the observation is an integer that denotes the document count at time interval t. That is, we assume the probability of generating an observation count O_t in state i is as follows:

$$B_i(O_t) = P(O_t|q_t = i) \sim N(u_i, \sigma_i)$$

The underlying states in our cHMM capture the burstiness of tweets during a particular time interval. A bursty state might correspond to a time when there are lots of users postings tweets (for example, when something newsworthy is taking place). A quiet state would correspond to times when nothing interesting is happening. In our current implementation, our cHMM uses three hidden states, but the model can be extended to capture arbitrarily many gradations of burstiness. The state transitions in our cHMM model sequential dependencies in these states—for example, a burst "dies down" when a newsworthy event passes. In each state, the mean u controls the "intensity" of the burst (i.e., how many documents are generated), and σ controls variations in different instances of the same state.

Figure 2 shows the three-state cHMM in our current implementation: circles represent states and arrows represent transitions. The blue circle denotes a "bursty" state as it has the largest mean, while the white circle can be interpreted as an "inactive" state since it has the smallest mean; the gray circle might be interpreted as an intermediate state.

Thus, our cHMM is parameterized as $\lambda = (A, u, \sigma, \pi)$. Given a sequence of observations (document counts within a fixed time window), we can derive an EM algorithm to estimate the parameters iteratively.

In the E-step, the expectation of the complete-data log-likelihood $\log P(O, q|\lambda)$, namely the Q *function*, is:

$$Q(\lambda, \lambda') \propto \sum_q \log P(O, q|\lambda) P(O, q|\lambda') \qquad (1)$$

where λ' represents estimates of parameters in the previous iteration that are known in the calculation and λ represents

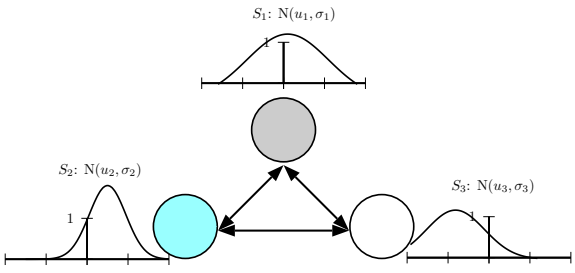

Figure 2: An illustration of a three-state cHMM. Each circle represents a state and arrows represent transitions. The Gaussians represent emissions (count of documents) from each state.

unknown parameters that we are trying to estimate for maximizing the Q function.

From the independence assumptions of HMMs (namely, that the observation O_t is only dependent on state q_t; state q_t is only dependent on the previous state q_{t-1}), we can compute the joint probability $P(O, q|\lambda)$ as follows:

$$\begin{aligned} P(O, q|\lambda) &= P(q|\lambda)P(O|q, \lambda) \\ &= \pi_{q_1} \prod_{t=2}^{T} A_{q_{t-1}q_t} \prod_{t=1}^{T} B_{q_t}(O_t) \end{aligned} \tag{2}$$

Substituting $P(O, q|\lambda)$ in Equation (2) into Equation (1):

$$\begin{aligned} Q(\lambda, \lambda') = &\sum_q \log \pi_{q_1} P(O, q|\lambda') \\ &+ \sum_q \left(\sum_{t=2}^{T} \log A_{q_{t-1}q_t} \right) P(O, q|\lambda') \\ &+ \sum_q \left(\sum_{t=1}^{T} \log B_{q_t}(O_t) \right) P(O, q|\lambda') \end{aligned} \tag{3}$$

We have broken the overall objective into three independent parts that we can optimize individually in the M-step. Since the optimization shares similarities with discrete HMMs (we recommend the tutorial by Bilmes [2] for more details), we skip the detailed derivations here and provide the final solutions as follows:

$$\pi_i = \frac{P(O, q_1 = i|\lambda')}{P(O|\lambda')} \tag{4}$$

$$A_{ij} = \frac{\sum_{t=2}^{T} P(O, q_{t-1} = i, q_t = j)}{\sum_{t=2}^{T} P(O, q_{t-1} = i|\lambda')} \tag{5}$$

$$u_i = \frac{\sum_{t=1}^{T} O_t \cdot P(O, q_t = i|\lambda')}{\sum_{t=1}^{T} P(O, q_t = i|\lambda')} \tag{6}$$

$$\sigma_i^2 = \frac{\sum_{t=1}^{T} (O_t - u_i)^2 \cdot P(O, q_t = i|\lambda')}{\sum_{t=1}^{T} P(O, q_t = i|\lambda')} \tag{7}$$

As with any EM algorithm, we iteratively update the parameters using above derivations until convergence. After arriving at the final parameter estimates $\lambda_f(A, u, \sigma, \pi)$, we can then use the Viterbi algorithm to find the sequence of states q_{opt} that maximizes $P(O|\lambda_f)$. Expansion terms are then computed from this state sequence, explained next.

3.2 Temporal Query Expansion

Given a query Q consisting of n query terms $\{t_1, t_2, ...t_n\}$, we use the above continuous hidden Markov model to find the state sequence that best describes the temporal distribution of the top k documents collected by an initial retrieval. We consider the state with the largest mean as the bursty state, and only select documents whose timestamps fall in the bursty state for query expansion. For convenience, we call these documents *bursty documents*. We then estimate a relevance model $P(w|R)$ [10] as follows:

$$P(w|R) = \sum_{D \in C} P(D)P(w|D) \prod_{i=1}^{n} P(t_i|D) \tag{8}$$

where C is the set of bursty documents. We assume uniform priors $P(D)$, so the relevance model is simply a weighted average of the terms in the documents, where the weights are the query likelihood scores.

Finally, just as in RM3, we interpolate the estimated relevance model with the original query model:

$$P'(w|R) = \alpha \cdot P(w|R) + (1 - \alpha) \cdot P(w|Q) \tag{9}$$

The interpolation parameter α is set to 0.5 by default. Following common parameter settings, we estimated the relevance models from $k = 50$ pseudo-relevant documents and selected $m = 20$ feedback terms.

4. EXPERIMENTS

We evaluated our model on the Tweets2011 corpus using test collections from the Microblog tracks at TREC 2011 and 2012. The Tweets2011 corpus contains ∼1% of all tweets from January 23, 2011 to February 7, 2011, totaling around 16 million tweets. There are 49 topics in TREC 2011 and 60 topics in TREC 2012, with relevance judgments assigned one of three grades: not relevant, relevant, and highly relevant. In our experiments, we considered both relevant and highly-relevant tweets as relevant. We removed retweets in the query expansion and final results, as the track guidelines consider them not relevant by fiat.

Our experimental procedure was as follows: We first performed initial retrieval using query-likelihood to gather ranked lists of tweets from the corpus. We then trained our cHMM on the top 50 tweets for each topic as described in Section 3.1, with three states and the number of time intervals T set to 30. After the cHMM parameters have been estimated via EM, we apply Viterbi decoding to extract the most likely state sequence, which is then used for temporal query expansion as described in Section 3.2. Note that this experimental procedure does not require a training/test split of the topics.

Our cHMM temporal pseudo-feedback technique is compared against the RM3 pseudo-feedback technique [10, 1] as a baseline. We also implemented the KDE variant of RM3 [7, 13], which includes four different ways to estimate feedback parameters: uniform, score-based, rank-based, and oracle. The first three are based on pseudo-feedback because they do not rely on user relevance judgments in the initial retrieved hits (which is the same as with RM3 and cHMM), while the oracle method requires explicit relevance

Method	P5	P15	P30	MAP
QL	0.465	0.411	0.354	0.268
RM3	0.500	0.433	0.378	0.302
RM3 + KDE (score)	0.494	0.436	0.379	0.300
RM3 + KDE (rank)	0.490	0.425	0.376	0.292
RM3 + KDE (oracle)	0.548$^\bullet$	0.492$^\bullet$	0.422$^\bullet$	0.319$^\bullet$
cHMM	**0.528**$^\bullet$	**0.444**$^\bullet$	**0.391**$^\circ$	**0.310**$^\circ$

Table 1: Experimental results comparing the effectiveness of cHMMs against RM3 and KDE variants.

judgments. The oracle, naturally, is not realistic, but is nevertheless useful to illustrate upper bound effectiveness. Here, we include results for the score-based, rank-based, and oracle conditions. We follow the same parameter tuning procedure in Rao et al. [13], where the parameters were learned using test data from TREC 2013 and 2014 topics. For completeness, we show the results of the initial query-likelihood (QL) retrieval without any feedback (this, of course, is a weak condition to compare against). In terms of evaluation metrics, we computed mean average precision (MAP) to 1000 hits and precision at ranks 5, 15 and 30 (P5, P15, P30), computed using `trec_eval`.

Experimental results are reported in Table 1. The symbols $\circ$ and $\bullet$ indicate that differences with respect to the RM3 baseline are statistically significant at $p < 0.10$ and $p < 0.05$ based on Fisher's two-sided, paired randomization test [14], respectively. We observe that QL is relatively ineffective as all other models outperform it by a large margin (all differences are statistically significant at $p < 0.01$). This replicates the robust finding that query expansion is effective for searching tweets.

Consistent with the findings in Efron et al. [13], the KDE (score) and KDE (rank) approaches do not improve upon the effectiveness of RM3 by itself. However, our cHMM approach significantly outperforms RM3, confirming our initial intuitions—we obtain higher-quality expansion terms from bursty documents, and that bursty states can be captured with our cHMM. The results of the KDE (oracle) condition are not surprising, since it exploits users' explicit relevance feedback. This condition can be viewed as an upper bound on how much temporal signal can be extracted to improve relevance ranking (at least with this broad class of techniques)—and results show that our cHMM achieves effectiveness that is pretty close to this upper bound.

As a specific example of how our cHMM helps, we took a closer look at topic 14 "release of The Rite", which achieves an improvement of 0.22 (MAP) and 0.57 (P30) against the RM3 baseline. We visualized the estimated cHMM state sequence from day 6 to day 1 in Figure 3. As there are too many states to show if we follow the setting of $T = 30$ in our experiments above, we reduced the number of states to one per day for illustrative purposes. The blue circle denotes a bursty state, the gray circles denote an intermediate (less bursty) state, and the white circle denotes an inactive state. As we can see, the bursty state reflects the cluster of documents at day 3 in the distribution of relevant documents (topic 14 in Figure 1). From day 6 to day 1, the inferred states reflect the density of the documents along the timeline. Overall, this example suggests that our cHMM is able to capture sequential dependencies in the temporal distribution of relevance, which is essential for identifying those bursty and expressive terms for expansion.

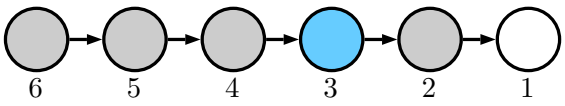

Figure 3: State evolution of topic 14 "release of The Rite" from day 6 to day 1 (each circle = one day). The blue circle represents a bursty state, the gray circles represent an intermediate state, and the white circle represents an inactive state.

5. CONCLUSIONS

This work contributes to a long thread of research on exploiting temporal signals for relevance ranking. We focus on query expansion via pseudo-relevance feedback, exploring the idea that expansion terms should be drawn from bursty documents. This is formalized via a continuous hidden Markov model to identify such documents, and our technique leads to significant effectiveness improvements over the popular RM3 model on standard TREC data. Our cHMM, however, represents only one possible model, and we are interested in exploring alternative approaches for capturing the temporal evolution of documents.

Acknowledgments. This work was supported by the U.S. National Science Foundation (NSF) under IIS-1218043 and CNS-1405688 and the Natural Sciences and Engineering Research Council of Canada (NSERC). Any opinions, findings, conclusions, or recommendations expressed are solely those of the authors.

6. REFERENCES

[1] N. Abdul-Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, D. Metzler, M. D. Smucker, T. Strohman, H. Turtle, and C. Wade. UMass at TREC 2004: Novelty and HARD. *TREC*, 2004.

[2] J. A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical report, International Computer Science Institute, 1998.

[3] J. Choi and W. B. Croft. Temporal models for microblogs. *CIKM*, 2012.

[4] O. Craveiro, J. Macedo, and H. Madeira. Query expansion with temporal segmented texts. *ECIR*, 2014.

[5] W. Dakka, L. Gravano, and P. G. Ipeirotis. Answering general time-sensitive queries. *TKDE*, 24(2):220–235, 2012.

[6] M. Efron and G. Golovchinsky. Estimation methods for ranking recent information. *SIGIR*, 2011.

[7] M. Efron, J. Lin, J. He, and A. de Vries. Temporal feedback for tweet search with non-parametric density estimation. *SIGIR*, 2014.

[8] M. Keikha, S. Gerani, and F. Crestani. TEMPER: A temporal relevance feedback method. *ECIR*, 2011.

[9] J. Kleinberg. Bursty and hierarchical structure in streams. *KDD*, 2003.

[10] V. Lavrenko and W. B. Croft. Relevance based language models. *SIGIR*, 2001.

[11] X. Li and W. B. Croft. Time-based language models. *CIKM*, 2003.

[12] M.-H. Peetz, E. Meij, M. de Rijke, and W. Weerkamp. Adaptive temporal query modeling. *ECIR*, 2012.

[13] J. Rao, J. Lin, and M. Efron. Reproducible experiments on lexical and temporal feedback for tweet search. *ECIR*, 2015.

[14] M. D. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. *CIKM*, 2007.