Time-Sensitive Language Modelling for Online Term Recurrence Prediction

Dell Zhang[1], Jinsong Lu[1], Robert Mao[2], and Jian-Yun Nie[3]

[1] Birkbeck, University of London
London WC1E 7HX, UK
dell.z@ieee.org, jingsong.lu@gmail.com
[2] Microsoft Corp.
Dublin, Ireland
robmao@microsoft.com
[3] University of Montreal
Quebec, H3C 3J7 Canada
nie@IRO.UMontreal.CA

Abstract. We address the problem of online term recurrence prediction: for a stream of terms, at each time point predict what term is going to recur next in the stream given the term occurrence history so far. It has many applications, for example, in Web search and social tagging. In this paper, we propose a time-sensitive language modelling approach to this problem that effectively combines term frequency and term recency information, and describe how this approach can be implemented efficiently by an online learning algorithm. Our experiments on a real-world Web query log dataset show significant improvements over standard language modelling.

1 Introduction

Consider a stream of terms[4] w_i with time stamp t_i: $(w_1, t_1), (w_2, t_2), \ldots$, where t_i monotonically increases. At each time point, we would like to predict what term is going to recur next in the stream, given the term occurrence history so far. This problem of *online term recurrence prediction* has many applications. For example, predicting the next query to be reused in a stream of queries is key to *query auto-completion* [1], *query suggestion* [2], *information re-finding* [3], and *result caching/prefetching* [4–7] in Web search engines, while predicting the next tag to be reused in a stream of tags is key to *tag auto-completion* [1] and *tag suggestion/recommendation* [8–11] in social tagging services.

The standard language modelling [12, 13] approach to this problem relies solely on term frequency information to find the most probable next term to recur. However, our experience tells us that the terms occurred recently should have a higher probability of recurring than those occurred a long time ago, due to

[4] Here terms are just language units which may contain one or more words, e.g., Web queries and social tags.

the common phenomena of burst and drift in user interests [14]. For example, if a user searched ‘baseball’ yesterday and ‘basketball’ one month ago on the Web, she is more likely to search ‘baseball’ rather than ‘basketball’ again today. Such valuable information of term recency has been overlooked in standard language modelling.

In this paper, we propose a *time-sensitive language modelling* approach to this problem. It can effectively combine term frequency and term recency information. Furthermore, it can be implemented efficiently by an *online learning* [15] algorithm. Our experiments on a real-world Web query log dataset show that it brings significant improvements over standard language modelling.

The rest of this paper is organised as follows. In Section 2, we present our time-sensitive language modelling approach to the problem of online term recurrence prediction, and explain how it combines term frequency and term recency information effectively. In Section 3, we describe an online learning algorithm to implement the above approach efficiently. In Section 4, we empirically evaluate our technique on a real-world Web query log dataset. In Section 5, we review related work. In Section 6, we discuss future work. In Section 7, we make conclusions.

2 Approach

The basic idea of our time-sensitive language modelling technique is that each occurrence of a term will contribute to its probability of recurring in the future, but the amount or weight of contribution decays over time according to a *kernel function* $k(t, t_o)$. Here we use the *exponential decay* function as the kernel.

$$k(t, t_o) = \begin{cases} \exp(-\lambda(t - t_o)) & \text{if } t \geq t_o \\ 0 & \text{if } t < t_o \end{cases} \quad ,$$

where $\lambda \geq 0$ is called the *decay constant*. The physical interpretation of this model is that one occurrence of term w at time t_o has an initial weight $W(t_o) = 1$ and it ‘evaporates’ at a rate proportional to its weight at that time $W(t)$: $dW/dt = -\lambda W$. The *mean lifetime*, i.e, the time needed for the initial weight to be reduced by a factor of e, is given by $\tau = 1/\lambda$.

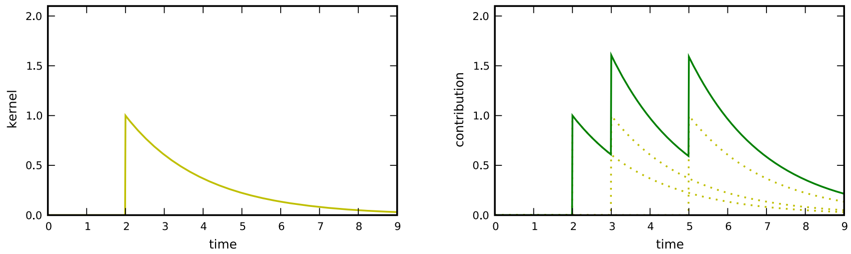
So at time t, if the history so far is $H = \{(w_1, t_1), \ldots, (w_n, t_n)\}$, then the accumulated weight of a specific term w's contribution can be calculated as

$$C(w, t) = \sum_{i=1}^{n} \delta(w, w_i) k(t, t_i) \ ,$$

where $\delta(w, w_i)$ is the Kronecker's delta function: $\delta(w, w') = 1$ if $w = w'$ and 0 otherwise. The function $C(w, t)$ summarises the contribution of term w's history to its occurring probability at time t, so we call it *contribution function*.

Figure 1 shows an example of time-sensitive language modelling based on exponential decay with $\lambda = 0.5$. In this example, the contribution of term w at time $t = 8$ will be

$$\begin{aligned} C(w, 8) &= k(8, 5) + k(8, 3) + k(8, 2) \\ &= \exp(-3\lambda) + \exp(-5\lambda) + \exp(-6\lambda) \\ &= \exp(-1.5) + \exp(-2.5) + \exp(-3) \\ &\approx 0.3550 \end{aligned}$$



(a) The kernel function $k(t, 2)$.

(b) The contribution function $C(w, t)$ for a term that occurred at times 2, 3 and 5.

Fig. 1. An example of time-sensitive language modelling based on exponential decay with $\lambda = 0.5$.

Hence the probability of term w occurring at time t, given the history H, will be determined by its accumulated contribution at that time:

$$\widehat{P}(w_{n+1} = w | H, t_{n+1} = t) = \widehat{P}(w | C(w, t)) = \frac{C(w, t) + \mu}{\sum_{w'} [C(w', t) + \mu]} \ .$$

where $0 \leq \mu \leq 1$ is a parameter for Lidstone (additive) smoothing [16]. If $\mu = 0$, the above formula gives the Maximum Likelihood Estimation (MLE) of w's occurring probability that combines both its frequency and its recency information. However, for a term never occurred before in the history (or the training corpus), the MLE of its occurring probability will be 0, which is in general undesirable. This problem can be remedied by using a positive smoothing parameter μ, which can be regarded as a non-decaying constant weight of contribution assigned to every term (whether occurred before or not) by default.

For the problem of online term recurrence prediction, what we need is to compute

$$w^* = \arg\max_{w} \widehat{P}(w_{n+1} = w | H, t_{n+1} = t) = \arg\max_{w} C(w, t) \ .$$

Now let's study the behaviour of time-sensitive language models in the context of online term recurrence prediction. Without loss of generality, we assume the system time is discrete, i.e., represented by an integer.

Proposition 1. *In a time-sensitive language model with decay constant $\lambda = 0$, the most probable next term to recur is the most frequently used (MFU) term.*

Proof. If $\lambda = 0$, then for each time point $t_i < t$ the kernel function $k(t, t_i) = exp(0) = 1$. So the contribution of term w at current time t is

$$C(w, t) = \sum_{i=1}^{n} \delta(w, w_i) k(t, t_i) = \sum_{i=1}^{n} \delta(w, w_i) \ ,$$

which is exactly the number of w's occurrences in the history. Therefore, the most probable next term $w^* = \arg\max_w C(w, t)$ is the term of highest frequency, i.e., the most frequently used (MFU) term. This also implies that when $\lambda = 0$, time-sensitive language modelling backs off to standard language modelling. □

Proposition 2. *In a time-sensitive language model with decay constant $\lambda \geq \ln(2)$, the most probable next term to recur is the most recently used (MRU) term.*

Proof. Let $r = \exp(-\lambda)$. Thus for each time point $t_i < t$ the kernel function can be written as $k(t, t_i) = \exp(-\lambda)^{t-t_i} = r^{t-t_i}$. Since $\lambda \geq \ln(2)$, we have $r \leq \exp(-\ln(2)) = \frac{1}{2}$. Consider two terms w_a and w_b that have occurred at times $T_a = \{t_{a_1}, \ldots, t_{a_n}\}$ and $T_b = \{t_{b_1}, \ldots, t_{b_n}\}$ respectively so far. Assume $t_{a_n} = u$ and $t_{b_n} = v$, and without loss of generality assume that $u < v$ or equivalently $u + 1 \leq v$ (as discrete time is used here).

$$\begin{aligned}
C(w_a, t) &= \sum_{i=1}^{n} \delta(w_a, w_i) k(t, t_i) = \sum_{t_i \in T_a} k(t, t_i) \\
&\leq \sum_{j=1}^{u} k(t, j) = \sum_{j=1}^{u} r^{t-j} = \frac{r^{t-(u+1)} - r^{t-1}}{1/r - 1} \\
&\leq r^{t-(u+1)} - r^{t-1} \\
&< r^{t-(u+1)} = k(t, u+1) \\
&\leq k(t, v) \ . \\
C(w_b, t) &= \sum_{i=1}^{n} \delta(w_b, w_i) k(t, t_i) = \sum_{t_i \in T_b} k(t, t_i) \\
&\geq k(t, v) \ .
\end{aligned}$$

To sum up, $C(w_a, t) < C(w_b, t)$ as long as $t_{a_n} < t_{b_n}$, i.e., the contribution of a term is dominated by its last occurrence time. Therefore, the most probable next term $w^* = \arg\max_w C(w, t)$ is the term of highest recency, i.e., the most recently used (MRU) term. □

The above two propositions show that at one extreme ($\lambda = 0$) the time-sensitive language modelling approach to online term recurrence prediction subsumes the approach of selecting the MFU terms (as in standard language modelling), while at the other extreme ($\lambda = \ln(2)$) it subsumes the approach of selecting the MRU terms. With a non-trivial decay constant $0 < \lambda < \ln(2)$, the time-sensitive language modelling approach differs from the MFU approach in that the contribution of each occurrence is not always the same but depends on its recency; it also differs from the MRU approach in that it considers not only the most recent occurrence, but also all the other past occurrences in the history. The decay constant λ controls the trade-off between term frequency and term recency in projecting a term's probability of being used in the future. What value of λ is optimal depends on the concrete problem and data.

3 Algorithm

A naive implementation of time-sensitive language modelling would be computationally expensive, because for each distinctive term, (1) we need to retain all its past occurrences for calculating its current contribution, and also (2) we need to constantly re-calculate its current contribution when time goes by. However, both issues turn out to be avoidable.

Theorem 1. *If there is no occurrence of term w in the period of time $[t_u, t_v]$, then its contribution satisfies*

$$C(w, t_v) = \exp(-\lambda(t_v - t_u))C(w, t_u) \ .$$

Proof. The contribution of term w at time t_u is

$$C(w, t_u) = \sum_{1 \le t_i \le t_u} \delta(w, w_i) k(t_u, t_i) \ ,$$

while the contribution of term w at time $t_v > t_u$ is

$$\begin{aligned} C(w, t_v) &= \sum_{1 \le t_i \le t_v} \delta(w, w_i) k(t_v, t_i) \\ &= \sum_{1 \le t_i \le t_u} \delta(w, w_i) k(t_v, t_i) + \sum_{t_u < t_i \le t_v} \delta(w, w_i) k(t_v, t_i) \ . \end{aligned}$$

As there is no occurrence of w between t_u and t_v, $\delta(w, w_i) = 0$ for all times t_i that $t_u < t_i \le t_v$. So we have

$$\begin{aligned} C(w, t_v) &= \sum_{1 \le t_i \le t_u} \delta(w, w_i) k(t_v, t_i) + \sum_{t_u < t_i \le t_v} 0 \cdot k(t_v, t_i) \\ &= \sum_{1 \le t_i \le t_u} \delta(w, w_i) k(t_v, t_i) \\ &= \sum_{1 \le t_i \le t_u} \delta(w, w_i) \exp(-\lambda(t_v - t_i)) \\ &= \sum_{1 \le t_i \le t_u} \delta(w, w_i) \exp(-\lambda((t_v - t_u) + (t_u - t_i)) \\ &= \exp(-\lambda(t_v - t_u)) \sum_{1 \le t_i \le t_u} \delta(w, w_i) \exp(\lambda(t_u - t_i)) \\ &= \exp(-\lambda(t_v - t_u))C(w, t_u) \ . \end{aligned}$$

□

Corollary 1. *The contribution of term w at the current time t can be calculated using its last occurrence (i.e., most recent occurrence) time t_l and its contribution at t_l:*

$$C(w,t) = \exp(-\lambda(t - t_l))C(w,t_l) \ .$$

Proof. Obviously there should be no occurrence of term w between its last occurrence time t_l and the current time t. So using the above theorem, we get this corollary straightforwardly. □

According to this corollary, we only need to retain two values, t_l and $C(w,t_l)$, for each distinctive term w in the system.

Corollary 2. *Suppose two different queries w_a and w_b last occurred at times t_{a_n} and t_{b_n} respectively. Let $t_m = \max(t_{a_n}, t_{b_n})$. At any time $t > t_m$ we have*

$$cmp(C(w_a,t), C(w_b,t)) = cmp(C(w_a,t_m), C(w_b,t_m)) \ ,$$

where cmp is the comparison function

$$cmp(x,x') = \begin{cases} +1 \ \mathit{if}\ x > x' \\ \ \ \ 0 \ \mathit{if}\ x = x' \\ -1 \ \mathit{if}\ x < x' \end{cases} \ .$$

Proof. Obviously there should be no occurrence of term w between the time t_m and the current time t. So using the above theorem, we get

$$C(w_a,t) = \exp(-\lambda(t - t_m))C(w_a,t_m)$$
$$C(w_b,t) = \exp(-\lambda(t - t_m))C(w_b,t_m) \ .$$

Hence their ratio $\frac{C(w_a,t)}{C(w_b,t)} = \frac{C(w_a,t_m)}{C(w_b,t_m)}$. So if $\frac{C(w_a,t_m)}{C(w_b,t_m)} > 1$, i.e., $C(w_a,t_m) > C(w_b,t_m)$, then $\frac{C(w_a,t)}{C(w_b,t)} > 1$, i.e., $C(w_a,t) > C(w_b,t)$. Similar conclusions can be drawn for the cases $\frac{C(w_a,t_m)}{C(w_b,t_m)} = 1$ and $\frac{C(w_a,t_m)}{C(w_b,t_m)} < 1$ as well. Thereby completing the proof. □

According to this corollary, although the contribution of each term changes over time, the relative order of two terms' contributions or their probabilities to occur next does not change, until either of them recurs. Therefore the only chance for another term to replace the current most probable next term w^* is when it occurs.

The above two corollaries enable us to cut the computational overhead of time-sensitive language modelling drastically. Furthermore, they make it possible to continuously update and apply a time-sensitive language model through *online learning* [15] so that it can deal with *stream data* [14].

Figure 2 shows the online term recurrence prediction algorithm using a time-sensitive language model. The *input* of this algorithm is a series of time-stamped

```
create a hash table h
$w^* = $ '␣'; h$[w^*] = \langle 0, 0 \rangle$
for $1 \le i \le n$:
    if $w_i$ not in h:
        $C(w_i, t_i) = 1$
    else:
        $\langle t_l, C(w_i, t_l) \rangle = $ h$[w_i]$
        $C(w_i, t_i) = \exp(-\lambda(t_i - t_l))C(w_i, t_l) + 1$
    h$[w_i] = \langle t_i, C(w_i, t_i) \rangle$
    $\langle t_l^*, C(w^*, t_l^*) \rangle = $ h$[w^*]$
    $C(w^*, t_i) = \exp(-\lambda(t_i - t_l^*))C(w^*, t_l^*)$
    if $C(w_i, t_i) > C(w^*, t_i)$:
        $w^* = w_i$
    output $w^*$
```

Fig. 2. The online term recurrence prediction algorithm using a time-sensitive language model.

term occurrences $\{(w_1, t_1), \ldots, (w_n, t_n)\}$ as well as the decay constant λ, and the *output* is a series of predictions for the most probable next term w^*. Hash table, a data structure that associates keys with values, has been used in the algorithm to store for each distinctive term w its last occurrence time t_l and contribution $C(w, t_l)$. A hash table supports lookup, insertion and deletion of elements in O(1) time (i.e., constant time) on average [17]. It is easy to see that overall this algorithm has time complexity O(n) for a stream of n terms.

4 Experiments

We apply the proposed technique of time-sensitive language modelling to *personalised query recurrence prediction* in Web search: considering each user's search log as a stream of queries, at each time point predict what query will be reused by her based on her entire query history so far. We assume that each user has her own time-clock and it is incremented by one upon each query issued.

We use the AOL query log dataset [18] (that is provided to the research community by AOL search engine[5]) for our experiments. In this paper, we focus on the queries within the first week of March 2006. The queries have already been normalised through punctuation-removal and case-folding etc. Finally the query log dataset used in this paper consists of $1,908,135$ queries from $309,078$ users.

One point worth mentioning is that in this dataset, if a user requested the next "page" of results for some query, this appears as a subsequent identical query with a later time stamp. Therefore it is not possible for us to determine

[5] http://search.aol.com/

whether the user reused the last query or she just requested for more results for the same query. To avoid this systematic bias towards the MRU approach in the experiments, we merge all identical successive queries from the same user into one. However, this implies that the immediate next query would always be different with the current one and the MRU approach would never succeed. To make a fair comparison between time-sensitive language modelling and standard language modelling, a small trick is applied here: we are actually making query recurrence predictions *two* steps forward, i.e., at time t_i we predict the query to be reused at time $t_i + 2$ by that user.

We evaluate the performance of our time-sensitive language modelling approach by comparing the top predicted queries with the later really occurred queries and computing the prediction accuracy. Here we report the accuracy of predicting recurred queries but not unseen queries, as the latter cannot be predicted based on the query history.

The experimental results for personalised query recurrence prediction using different decay constants are shown in Figure 3 and Table 1. The time-sensitive language modelling approach significantly outperforms the standard language modelling (i.e., MFU) approach. By using a non-zero decay constant to combine query frequency and query recency, we get an accuracy improvement of *more than a third*. The optimal decay constant $\lambda^* \approx 0.2 \times \ln 2 = 0.1386$. According to one-sided *s*-test [19, 20], the effectiveness superiority of time-sensitive language modelling over standard language modelling is at the significance level 99.9% (P_value < 0.001).

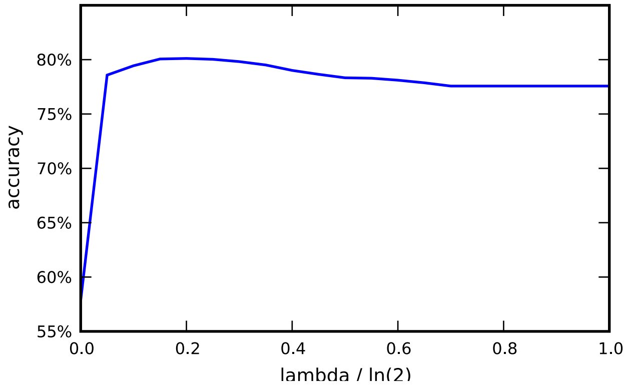


Fig. 3. The experimental results for personalised query recurrence prediction.

To analyse the underlying reason for the success of time-sensitive language modelling, we rank all distinctive query recurrence incidents according to their frequency or recency values (in the corresponding user's individual query history) and calculate the proportion of query recurrence incidents for each frequency or recency rank. Figure 4 shows the *log-log* plots of query recurrence proportion

Table 1. The experimental results for personalised query recurrence prediction.

language model	decay constant	prediction accuracy
standard	$\lambda = 0.0 \times \ln 2$ (MFU)	57.86%
time-sensitive	$\lambda = 0.1 \times \ln 2$	79.44%
	$\lambda = 0.2 \times \ln 2$	**80.11%**
	$\lambda = 0.3 \times \ln 2$	79.82%
	$\lambda = 0.4 \times \ln 2$	79.01%
	$\lambda = 0.5 \times \ln 2$	78.33%
	$\lambda = 0.6 \times \ln 2$	78.11%
	$\lambda = 0.7 \times \ln 2$	77.57%
	$\lambda = 0.8 \times \ln 2$	77.57%
	$\lambda = 0.9 \times \ln 2$	77.57%
	$\lambda = 1.0 \times \ln 2$ (MRU)	77.57%

over its frequency and recency rank respectively. We observe that in general, consistent with our intuition, (1) more *frequently* used queries are more likely to recur; and (2) more *recently* used queries are more likely to recur. We compute Kendall's rank-correlation coefficient τ [21] to quantitatively measure the utility of query frequency and query recency in predicting query recurrence. The value of τ for the former is 0.4470 while that for the latter is 0.9367, which means that query recurrence is much more dependent on query recency than query frequency. Therefore the valuable information of query recency must not be discarded for personalised query recurrence prediction.

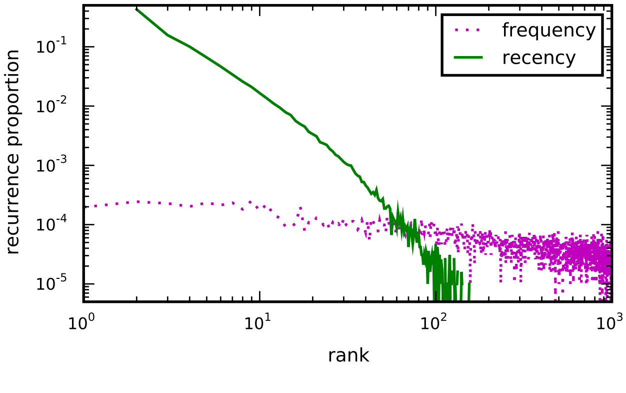


Fig. 4. The proportion of query recurrence over query frequency/recency.

5 Related Work

The idea of making language models adaptive by introducing a decay function has appeared in various contexts such as speech recognition [22], news retrieval [23], email clustering [24], and collaborative filtering [25]. However, to the best of our knowledge, the effective behaviour and efficient implementation of time-sensitive language modelling for the problem of online term recurrence prediction have not been studied before.

6 Future Work

In this paper, we have focused on incorporating temporal decay into unigram language modelling, but it should be straightforward to extend our proposed technique to general n-gram language modelling [12, 13] or even topic modelling [26]. It will also be interesting to investigate other temporal kernel functions in the proposed framework of time-sensitive language modelling, e.g., a periodic one to model the pattern of repetitive term occurrences.

7 Conclusions

The major contribution of this paper is the technique of time-sensitive language modelling that can address the problem of online term recurrence prediction effectively and efficiently.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments.

References

1. Garay-Vitoria, N., Abascal, J.: Text prediction systems: A survey. Universal Access in the Information Society **4**(3) (2006) 188–203
2. Mei, Q., Zhou, D., Church, K.W.: Query suggestion using hitting time. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM), Napa Valley, CA, USA (2008) 469–478
3. Teevan, J., Adar, E., Jones, R., Potts, M.A.S.: Information re-retrieval: Repeat queries in yahoo's logs. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Amsterdam, The Netherlands (2007) 151–158
4. Lempel, R., Moran, S.: Predictive caching and prefetching of query results in search engines. In: Proceedings of the 12th International World Wide Web Conference (WWW), Budapest, Hungary (2003) 19–28
5. Baeza-Yates, R.A., Gionis, A., Junqueira, F., Murdock, V., Plachouras, V., Silvestri, F.: The impact of caching on search engines. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Amsterdam, The Netherlands (2007) 183–190

6. Fagni, T., Perego, R., Silvestri, F., Orlando, S.: Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data. ACM Transactions on Information Systems (TOIS) **24**(1) (2006) 51–78
7. Gan, Q., Suel, T.: Improved techniques for result caching in web search engines. In: Proceedings of the 18th International Conference on World Wide Web (WWW), Madrid, Spain (2009) 431–440
8. Heymann, P., Ramage, D., Garcia-Molina, H.: Social tag prediction. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Singapore (2008) 531–538
9. Sigurbjornsson, B., van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: Proceedings of the 17th International World Wide Web Conference (WWW), Beijing, China (2008) 327–336
10. Song, Y., Zhuang, Z., Li, H., Zhao, Q., Li, J., Lee, W.C., Giles, C.L.: Real-time automatic tag recommendation. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Singapore (2008) 515–522
11. Song, Y., 0007, L.Z., Giles, C.L.: A sparse gaussian processes classification framework for fast tag suggestions. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM), Napa Valley, CA, USA (2008) 93–102
12. Manning, C., Schutze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA (1999)
13. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
14. Gaber, M.M., Zaslavsky, A.B., Krishnaswamy, S.: Mining data streams: A review. SIGMOD Record **34**(2) (2005) 18–26
15. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer-Verlag (2006)
16. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. In: Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL), Morristown, NJ, USA, Association for Computational Linguistics (1996) 310–318
17. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. 2nd edn. MIT Press and McGraw-Hill (2001)
18. Pass, G., Chowdhury, A., Torgeson, C.: A picture of search. In: Proceedings of the 1st International Conference on Scalable Information Systems (Infoscale), Hong Kong (2006) 1
19. Mitchell, T.: Machine Learning. international edn. McGraw Hill (1997)
20. Yang, Y., Liu, X.: A re-examination of text categorization methods. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), Berkeley, CA (1999) 42–49
21. Kendall, M., Gibbons, J.D.: Rank Correlation Methods. 5th edn. A Charles Griffin Book (1990)
22. Clarkson, P.R., Robinson, A.J.: Language model adaptation using mixtures and an exponentially decaying cache. In: Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Volume 2. (1997) 799–802
23. Li, X., Croft, W.B.: Time-based language models. In: Proceedings of the 12th ACM Conference on Information and Knowledge Management (CIKM), New Orleans, LA, USA (2003) 469–475
24. Zhu, X., Ghahramani, Z., Lafferty, J.: Time-sensitive dirichlet process mixture models. Technical Report CMU-CALD-05-104, Carnegie Mellon University (2005)

25. Ding, Y., Li, X.: Time weight collaborative filtering. In: CIKM, Bremen, Germany (2005) 485–492
26. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research (JMLR) **3** (2003) 993–1022