Using Temporal Language Models for Document Dating

Nattiya Kanhabua and Kjetil Nørvåg

Dept. of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway

Abstract. In order to increase precision in searching for web pages or web documents, taking the temporal dimension into account is gaining increased interest. A particular problem for web documents found on the Internet is that in general, no trustworthy timestamp is available. This is due to its decentralized nature and the lack of standards for time and date. In previous work we have presented techniques for solving this problem. In this paper, we present a tool for determining the timestamp of a non-timestamped document (using file, URL or text as input) using temporal language models. We also outline how this tool will be demonstrated.

1 Introduction

In order to increase precision in searching for web pages or web documents, taking the temporal dimension into account is gaining increased interest. In this way, the search engine will retrieve documents according to both text and temporal criteria, i.e., *temporal text-containment search* [5].

Due to its decentralized nature and the lack of standards for time and date, it is difficult to determine an accurate and trustworthy timestamp of a web document. In a web warehouse or a web archive, there is no guarantee that the creation time and the time of retrieval by the crawler are related.

In this paper, we present a tool for determining timestamp of a non-timestamped document using temporal language models. The tool can take as input a file, contents from an URL, or text entered directly. As output it will present an estimation of possible creation time/periods, with confidence of each of the estimated time periods. Obviously, the one with highest confidence is the most probable based on the language model. An example of the interface is shown in Fig. 1(a) and example of results are shown in Fig. 1(b-e).

To build a system for dating a document, we compare document contents with word statistics and usages over time. The dating approach is based on the *temporal language model* presented in [1]. The intuition behind this approach is that, for a given document with unknown timestamp, it is possible to find the time partition that mostly overlaps in term usage with the document. For example, if the document contains the word "tsunami" and corpus statistics shows this word was very frequently used in 2004/2005, it can be assumed that this time period is a good candidate for the document timestamp. The model assigns a probability to a document according to word statistics over time. By partitioning a document corpus into time partitions, it is possible to determine the

timestamp of a non-timestamped document d_i by computing a similarity score (*NLLR*) between the language model of d_i with each partition p_j. The timestamp of the document is the partition which maximizes the similarity score.

The rest of the paper is organized as follows. In Sect. 2 we outline the temporal language models used in our approach. In Sect. 3 we describe our document dating prototype. Finally, in Sect. 4 we outline our proposed demo.

2 Temporal Language Models

Timestamp estimation is based on the statistic language model presented by de Jong, Rode and Hiemstra [1]. This *temporal language model* is a variant of the time-based model in [4], based on a probabilistic model from [6]. The temporal language model assigns a probability to a time partition according to word usage or word statistics over time.

A document is modeled as $d_i = \{\{w_1, \ldots, w_n\}, (t_i, t_{i+1})\}$ where $t_i < t_{i+1}$ and (t_i, t_{i+1}) is a temporal view of document which can be represented by a time partition associated to its timestamp. A normalized log-likelihood ratio [3] is used to compute the similarity between two language models. Given a partitioned corpus, it is possible to determine the timestamp of a non-timestamped document d_i by comparing the language model of d_i with each corpus partition p_j using the following equation:

$$Score(d_i, p_j) = \sum_{w \in d_i} P(w|d_i) \times \log \frac{P(w|p_j)}{P(w|C)} \tag{1}$$

where C is the background model estimated on the entire collection and p_j is a time partition. The timestamp of the document is the partition maximizing a score according to the equation above, and the confidence *Conf* of the estimation is calculated as the logarithm of the score of the highest ranked relative to the second ranked partition.

In [2] we presented improvements to the approach of [1], the most important being temporal entropy, use of search statistics and adapted semantic-based preprocessing.

We use *temporal entropy* (TE) to weight terms differently depending on how well a term is suitable for separating time partitions among overall time partitions and also indicates how important a term is in a specific time partition. Temporal entropy of a term w_i is given as follows:

$$TE(w_i) = 1 + \frac{1}{\log N_P} \sum_{p \in \mathbf{P}} P(p|w_i) \times \log P(p|w_i) \tag{2}$$

where $P(p_j|w_i) = \frac{tf(w_i,p_j)}{\sum_{k=1}^{N_P} tf(w_i,p_k)}$, N_P is the total number of partitions in a corpus $\mathbf{P}$, and $tf(w_i, p_j)$ is the frequency of w_i in partition p_j. Modifying the score in Equation (1), each term w can be weighted with temporal entropy $TE(w)$ as follows:

$$Score_{te}(d_i, p_j) = \sum_{w \in d_i} TE(w) \times P(w|d_i) \times \log \frac{P(w|p_j)}{P(w|C)} \tag{3}$$

Search statistics provided by Google Zeitgeist (GZ) can be integrated as an additional score in order to increase the probability of a tentative time partition. GZ essentially gives statistics of trends of search terms, i.e., increasing and decreasing popularity. By analyzing search statistics, we are able to increase the probability for a particular partition which contains top-ranked queries. The higher probability the partition acquires, the more potential time candidate it becomes. *GZ* is integrated as an additional score into Equation (1) in order to increase the probability of partition p_j:

$$Score_{gz}(d_i, p_j) = \sum_{w \in d_i} \left(P(w|p_j) \times \log \frac{P(w|p_j)}{P(w|C)} + \beta GZ(p_j, w) \right) \quad (4)$$

where β is the weight for the *GZ* function (see [2] for more details on calculating GZ).

In order to further increase accuracy of the dating, we have also integrated *semantic-based techniques* into document preprocessing, i.e., part-of-speech tagging (POS), collocation extraction (COLL), word sense disambiguation (WSD), and concept extraction (CON).

3 Document Dating System

Our prototype implements the ideas from [2], and uses a web-based interface. It allows to estimate the date of different input formats (i.e., a file, an URL, or plain text) as shown by Fig. 1(a). Example inputs can be URL: "http://tsunami-thailand.blogspot.com" or text: "the president Obama". The user can select parameters: preprocessing (POS, COLL, WSD, or CON), similarity score (NLLR, GZ or TE), and time granularity (1-month, 3-months, 6-months, or 12-months). Given an input to be dated, the system computes similarity scores between a given document/text and temporal language models. The document is then associated with tentative time partitions or its likely originated timestamps. The results can be displayed in two ways. First, a rank list of partitions is shown in an descending order according to their scores. Second, each tentative time partition is drawn in a timeline with its score as a height.

4 Demo Outline

In the demo, we will present the features of our dating tool, including the impact of the variants of our temporal language approach:

Basic vs. advanced preprocessing: There is a trade-off among semantic-based preprocessing. We compare a *basic* preprocessing (POS only) to an *advanced* preprocessing (a combination of POS, COLL, WSD, and CON). As will be shown, *basic* used less time, but gains a poorer quality than the *advanced*.

How GZ enhances scores: To improve the accuracy, we compute scores by using *GZ* in addition to *NLLR*. The correct time period (2004/12 to 2005/11) is raised from the 7^{th} rank in Fig. 1(b) to the 1^{st} rank with higher confidence in Fig. 1(c).

TE as a trend: A term occurring in few partitions is weighted high by *TE* and it provides high scores for partitions in which the term appears. Fig. 1(d-e) display trends of the web page about "US presidential election" with and without *TE* respectively and *TE* gives higher scores for relevant periods (2000, 2004 and 2008).

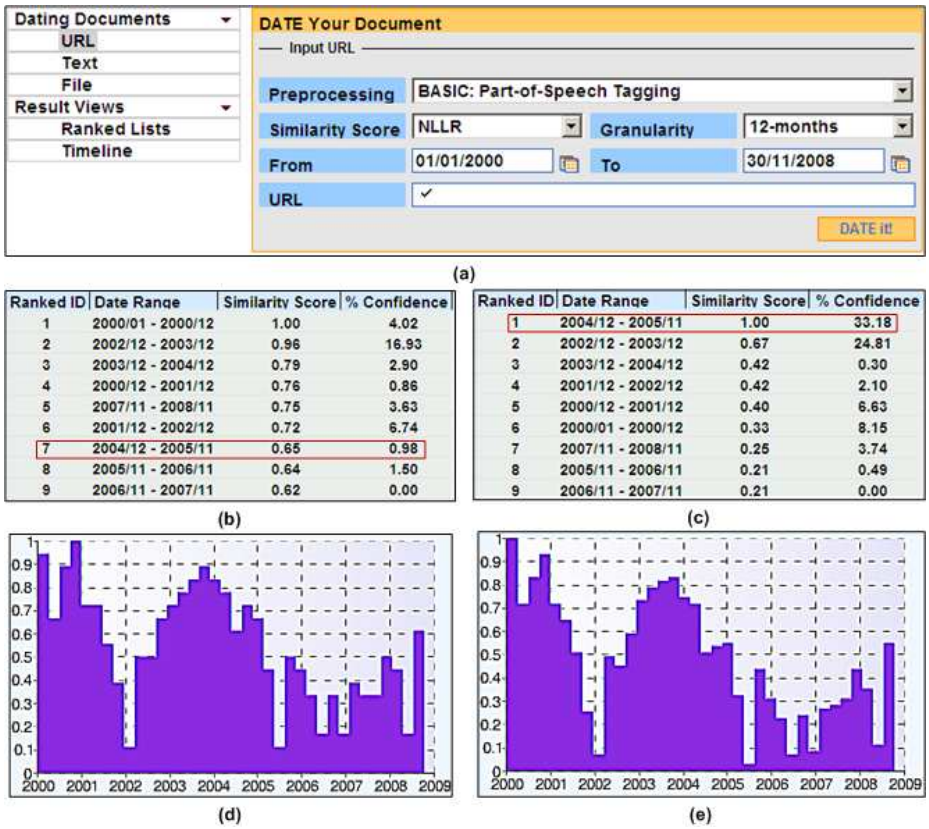

Fig. 1. (a) System interface, (b) Results of *basic* preprocessing and *NLLR*, (c) Results of *basic* preprocessing and *GZ*, (d-e) Trends of "US presidential election" with and without *TE*.

References

1. F. de Jong, H. Rode, and D. Hiemstra. Temporal language models for the disclosure of historical text. In *Proceedings of AHC'2005 (History and Computing)*, 2005.
2. N. Kanhabua and K. Nørvåg. Improving temporal language models for determining time of non-timestamped documents. In *Proceedings of ECDL'2008*, 2008.
3. W. Kraaij. Variations on language modeling for information retrieval. *SIGIR Forum*, 39(1):61, 2005.
4. X. Li and W. B. Croft. Time-based language models. In *Proceedings of CIKM'2003*, 2003.
5. K. Nørvåg. Supporting temporal text-containment queries in temporal document databases. *Journal of Data & Knowledge Engineering*, 49(1):105–125, 2004.
6. J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of SIGIR'1998*, 1998.