Event-based Time Label Propagation for Automatic Dating of News Articles

Tao Ge **Baobao Chang**[*] **Sujian Li** **Zhifang Sui**
Key Laboratory of Computational Linguistics, Ministry of Education
School of Electronics Engineering and Computer Science, Peking University
No.5 Yiheyuan Road, Haidian District, Beijing, P.R.China, 100871
{getao,chbb,lisujian,szf}@pku.edu.cn

Abstract

Since many applications such as timeline summaries and temporal IR involving temporal analysis rely on document timestamps, the task of automatic dating of documents has been increasingly important. Instead of using feature-based methods as conventional models, our method attempts to date documents in a year level by exploiting relative temporal relations between documents and events, which are very effective for dating documents. Based on this intuition, we proposed an event-based time label propagation model called confidence boosting in which time label information can be propagated between documents and events on a bipartite graph. The experiments show that our event-based propagation model can predict document timestamps in high accuracy and the model combined with a MaxEnt classifier outperforms the state-of-the-art method for this task especially when the size of the training set is small.

1 Introduction

Time is an important dimension of any information space and can be useful in information retrieval, question-answering systems and timeline summaries. In the applications involving temporal analysis, document timestamps are very useful. For instance, temporal information retrieval models take into consideration the document's creation time for document retrieval and ranking (Kalczynski and Chou, 2005; Berberich et al., 2007) for better dealing with time-sensitive queries; some information retrieval applications such as Google Scholar can list articles published during the time a user specifies for better satisfying users' needs. In addition, timeline summarization techniques (Hu et al., 2011; Binh Tran et al., 2013) and some event-event ordering models (Chambers and Jurafsky, 2008; Yoshikawa et al., 2009) also rely on the timestamps. Unfortunately, many documents on the web do not have a credible timestamp, as Chambers (2012) reported. Therefore, it is significant to date documents, that is to predict document creation time.

One typical method for dating document is based on temporal language models, which were first used for dating by de Jong et al. (2005). They learned language models (unigram) for specific time periods and scored articles with normalized log-likelihood ratio scores. The other typical approach for the task was proposed by Nathanael Chambers (2012). In Chambers's work, discriminative classifiers – maximum entropy (MaxEnt) classifiers were used by incorporating linguistic features and temporal constraints for training, which outperforms the previous temporal language models on a subset of Gigaword Corpus (Graff et al., 2003).

However, the conventional methods have some limitations because they predict creation time of documents mainly based on feature-based models without understanding content of documents, which may lead to wrong predictions in some cases. For instance, assume that $D1$ and $D2$ are documents whose content is given as follows:

> ($D1$) Sudan *last year* accused Eritrea of backing an offensive by rebels in the eastern border region.

[*] Corresponding author

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1–11,
Seattle, Washington, USA, 18-21 October 2013.

($D2$) *Two years ago*, Sudan accused Eritrea of backing an offensive by rebels in the eastern border region.

Since $D1$ and $D2$ share many important features, the previous dating methods are very likely to predict the same timestamp for the two documents. However, it will be easy to infer that the creation time of $D1$ should be one year earlier than that of $D2$ if we analyze the content of the two documents.

Unlike the previous methods, this paper exploits relative temporal relations between events and documents for dating documents on the basis of an understanding of document content.

It is known that each event in a news article has a relative temporal relation with the document. By analyzing the relative temporal relation, time of the event can be known if we know the document timestamp; on the other hand, if the time of an event is known, it can also be used to predict the creation time of documents mentioning the event, which can be best demonstrated with the above-mentioned example of $D1$ and $D2$. In the example, "*last year*" is an important cue to infer that the event mentioned by the documents occurred in 2002 if we know the timestamp of $D1$ is 2003. With the information that the event occurred in 2002, it can also be inferred from the temporal expression "*Two years ago*" that $D2$ was written in 2004. In this way, the timestamp of the labeled document ($D1$) is propagated to the unlabeled document ($D2$) through the event both of them mention, which is the main intuition of this paper.

In fact, this intuition seems practical to date documents on the web because web data is very redundant. Many documents on the web can be connected via events because an event is usually mentioned by different documents. According to our analysis of a collection of news articles spanning 5 years, it is found that an event is mentioned by 3.44 news articles on average; on the other hand, a document usually refers to multiple events. Therefore, if one knows a document timestamp, time of events the document mentions can be obtained by analyzing the relative temporal relations between the document and the events. Likewise, if the time of an event is known, then it can be used to predict creation time of the documents which mention it.

Based on the intuition, we proposed an event-based time label propagation model called confidence boosting in which timestamps are propagated according to relative temporal relations between documents and events. In this way, documents can be dated with an understanding of content so that this model can date document more credibly. To our knowledge, it is the first time that the relative temporal relations between documents and events are exploited for dating documents, which is proved to be effective by the experimental results.

2 Event-based Time Label Propogation

As mentioned above, the relative temporal relations between documents and events are useful for dating documents. By analyzing the temporal relations, even if there are only a small number of documents labeled with timestamps, this information can be propagated to documents connected with them on a bipartite graph using breadth first traversal (BFS).

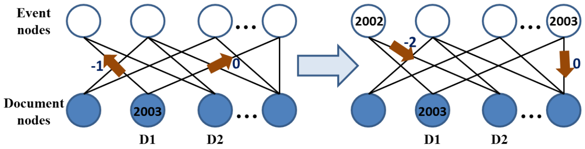

Figure 1: An example of BFS-based propagation

As shown in figure 1, there are two kinds of nodes in the bipartite graph. A document node is a single document while an event node represents an event. The edge between a document node and an event node means that the document mentions the event. Also, the edge carries the information of the relative temporal relation between the document and the event. The label propagation from node i to node j will occur if BFS condition which is defined as follows is satisfied:

$$\begin{cases} e_{ij} \in E \\ i \in L \text{ and } j \notin L \end{cases} \text{(BFS condition)}$$

When the timestamp of i is propagated to j:

$$Y(j) = Y(i) + \delta(i,j)$$
$$L = L \cup \{j\}$$

where E is the set of edges of the bipartite graph, e_{ij} denotes the edge between node i and j, L is the set of nodes which have been already labeled with timestamps, $Y(i)$ is the year of node i and $\delta(i,j)$ is the relative temporal relation between node i and j.

In figure 1, the timestamp of document $D1$ is 2003, which is known. This information can be propagated to its adjacent nodes i.e. the event nodes it mentions according to the relative temporal relations. Then, these event nodes propagate their timestamps to other documents which mention them. By repeating this process, the timestamp of the document can be propagated to documents which are reachable from the initially labeled document on the bipartite graph.

Although the BFS-based propagation process can propagate timestamps from few labeled documents to a large number of unlabeled ones, it has two shortcomings for this task. First, once one timestamp is propagated incorrectly, this error will lead to more mistakes in the following propagations. If such an error occurred at the beginning of the propagation process, it would lead to propagation of errors. Second, BFS-based method cannot address conflict of predictions during propagation, which is shown in figure 2.

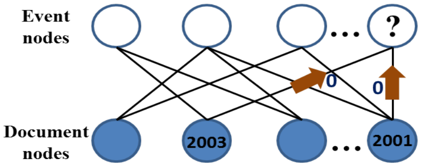

Figure 2: Conflict of predictions during propagation

To address the problems of the BFS-based method, we proposed a novel propagation model called confidence boosting model which improves the BFS-based model by optimizing the global confidence of the bipartite graph. In the confidence boosting model, every node in the bipartite graph has a confidence which measures the credibility of the predicted timestamp of the node. When the timestamp of a node is propagated to other nodes, its confidence will be also propagated to the target nodes with some loss. The loss of confidence is called confidence decay. Formally, the confidence decay process is described as follows:

$$c(j) = c(i) \times \sigma(i, j)$$

where $c(i)$ denotes confidence of node i and $\sigma(i, j)$ is the decay factor from node i to node j. For guaranteeing that timestamps can be propagated on the bipartite graph credibly, we define the following condition which is called CB (Confidence Boosting) condition:

$$\begin{cases} e_{ij} \in E \\ c(i) \times \sigma(i, j) > c(j) \end{cases} \text{(CB condition)}$$

In the confidence boosting model, propagation from node i to node j will occur only if CB condition is satisfied. When timestamps are propagated on the bipartite graph, timestamps and confidence of nodes will be updated dynamically. A node with high confidence is more active than nodes with low confidence to propagate its timestamp because a node with high confidence is more likely to satisfy the CB condition for propagating its timestamp. Moreover, a prediction with low confidence can be corrected by the prediction with high confidence. Therefore, the confidence boosting model can address both propagation of errors and conflict of predictions which cannot be tackled by the BFS-based model.

However, there are challenges for running such propagation models in practice. First, the relative temporal relations between documents and events are usually unavailable. Second, events extracted from different documents do not have any connection even if they refer to the same event. Therefore, each event is connected with only one document in the bipartite graph and thus cannot propagate its timestamp to other documents unless we perform event coreference resolution. Third, propagations from generic events are very likely to lead to propagation errors because generic events can happen in any year. Also, how to set the confidence and decay factors reasonably in practice for a confidence boosting model is worthy of investigation. All these challenges for the propagation models and their corresponding solutions will be discussed in Section 3.

3 Details of Event-based Propagation Models

In this section, details of the event-based time label propagation models including challenges and their corresponding solutions are presented. We first discuss the event extraction and processing involving relative temporal relation mining, event coreference resolution and distinguishing specific extractions from generic ones in Section 3.1. Then, we show the confidence boosting algorithm in detail in Section 3.2.

3.1 Event extraction and processing

As mentioned in previous sections, events play a key role in the propagation models. We define an event as a Subject-Predicate-Object (SPO) triple. To extract events from raw text, an open information extraction software - ReVerb (Fader et al., 2011) is used. ReVerb is a program that automatically identifies and extracts relationships from English sentences. It takes raw text as input and outputs SPO triples which are called extractions.

However, extractions extracted by ReVerb cannot be used directly for our propagation models for three main reasons. First, the relative temporal relations between documents and the extractions are unavailable. Second, the extractions extracted from different documents do not have any connection even if they refer to the same event. Third, propagations from generic events are very likely to lead to propagation errors.

For addressing the three challenges for the propagation models, we first presented a rule-based method for mining the relative temporal relations between extractions and documents in Section 3.1.1. Then, an efficient event coreference resolution method is introduced in Section 3.1.2. Finally, the method for distinguishing specific extractions from generic ones is shown in Section 3.1.3.

3.1.1 Relative temporal relation mining

We used a rule-based method to extract temporal expressions and used Stanford parser (De Marneffe et al., 2006) to analyze association between the temporal expressions and the extractions. Specifically, we define that an extraction is associated with a temporal expression if there is an arc from the predicate of the extraction to the temporal expression in the dependency tree. For a certain extraction, there are the following four cases whose instances are shown in table 1 for handling.

Case 1: The extraction is associated with an absolute temporal expressions with year mentions in the sentence.

In this case, the time of the extraction is equal to the year mention:

$$Y(ex) = YearMention$$

For the example in table 1, $Y(ex) = 1999$.

Case 2: The extraction is associated with a relative temporal expression (not involving year) in the sentence.

Case	Instance
1	*In 1999*, South Korea exported 89,000 tons of pork to Japan.
2	*In April*, however, the BOI investments showed marked improvement.
	Last month, Kazini vowed to resign his top army job.
3	Julius Erving moved with his family to Florida *three years ago*.
4	The meeting focused on ways to revive the stalled Mideast peace process.

Table 1: Instances of various temporal expressions

In this case, the time of the extraction is equal to the creation time of the document:

$$Y(ex) = Y(d)$$

Case 3: The extraction is associated with a relative temporal expression (involving specific year gap) in the sentence.

In this case, the time of the extraction is computed as follows:

$$Y(ex) = Y(d) \pm YearGap$$

For the example in table 1, $Y(ex) = Y(d) - 3$.

Case 4: The extraction is not associated with any temporal expression in the sentence or the other cases.

In this case, it is difficult to recognize the relative temporal relations. However, timeliness can be leveraged to determine the relations as a heuristic method. It is known that timeliness is an important feature of news so that events reported by a news article usually took place a couple of days or weeks before the article was written. Therefore, we heuristically consider the year of the extraction is the same with that of its source document in this case:

$$Y(ex) = Y(d)$$

In the cases except case 1, the relative temporal relation between an extraction and the document it comes from can be determined. To evaluate the performance of the rule-based method, we sampled 3,000 extractions from documents written in the year of 1995-1999 of Gigaword corpus and manually labeled these extractions with a timestamp based on their context and their corresponding document timestamps as golden standard. Table 2 shows the accuracy of each case which will be used as a part of the decay factor in the confidence boosting model.

Case	Accuracy
1	0.774(168/217)
2	0.994(844/849)
3	0.836(281/336)
4	0.861(1376/1598)
Total	0.890(2669/3000)

Table 2: Accuracy of the four cases

We define the set of these determined relative temporal relations R as follows:

$$R = \{r_{d,ex} | d = doc(ex), ex \in C_2 \cup C_3 \cup C_4\}$$
$$r_{d,ex} =< d, ex, \delta(d, ex) >$$
$$\delta(d, ex) = -\delta(ex, d) = \{0, \pm 1, \pm 2, \pm 3, ...\}$$

where C_k is the set of extractions in case k and $doc(ex)$ is the document which extraction ex comes from. $r_{d,ex}$ is a triple describing the relative temporal relation between d and ex. For example, triple $r_{d,ex} =< d, ex, -1 >$ means that the time of extraction ex is one year before the time of document d.

3.1.2 Event coreference resolution

Extractions from different documents have no connections. However, there are a great number of extractions referring to the same event. For finding such coreferential event extractions efficiently, hierarchical agglomerative clustering (HAC) is used to cluster highly similar extractions into one cluster. We use cosine to measure the similarity between extractions and select bag of words as features. Note that it is less meaningful to cluster the extractions from the same document because coreferential extractions from the same document are not helpful for timestamp propagations. For this reason, similarity between extractions from the same documents is set to 0.

For HAC, selection of threshold is important. If the threshold is set too high, only a few extractions can be clustered despite high purity; on the contrary, if the threshold is set too low, purity of clusters will descend. In fact, selection of threshold is a trade-off between the precision and recall of event coreference resolution. For selecting a suitable threshold, extractions from documents written in 1995-1999 are used as a development set.

In practice, it is difficult for us to directly evaluate the performance of the coreference resolution of event extractions without golden standard which requires much labors for manual annotations. Alternatively, entropy which measures the purity of clusters is used for evaluation because it can indirectly reflect the precision of coreference resolution to some extent:

$$\text{Entropy} = -\sum_j \frac{n_j}{n} \sum_i P(i,j) \times \log_2 P(i,j)$$

where $P(i, j)$ is the probability of finding an extraction whose timestamp is i in the cluster j, n_j is the number of items in cluster j and n is the total number of extractions. Note that timestamp of an extraction is assigned based on its document timestamp using the method proposed in Section 3.1.1.

Figure 3 shows the effect of selection of the threshold on cluster performance. It can be found that when the threshold reaches 0.8, the entropy starts descending gently and is low enough. Since we want to find as many coreferential extractions as possible on the premise that the precision is good, the threshold is set to 0.8. Note that extractions which are single in one cluster will be filtered out because they do not have any connections with any other documents.

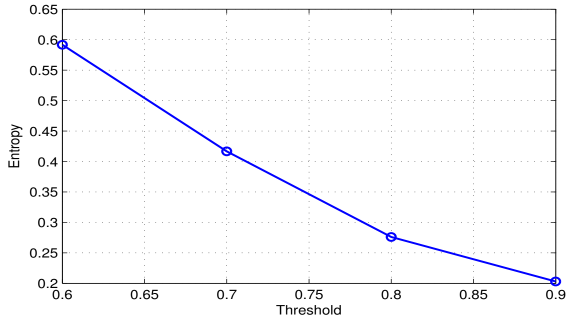

Figure 3: Entropy of clusters under different thresholds

3.1.3 Distinguishing specific events from generic ones

Not all extractions extracted by ReVerb refer to a specific event. For instance, the extraction "Germany's DAX index was down 0.2 percent" is undesirable for our task because it refers to a generic event and this event may occur in any year. In other words, it is not able to indicate a certain timestamp and thus propagations from a generic event node are very likely to result in propagation errors. In contrast, the extraction "some of the provinces in China were hit by SARS" refers to a specific event which took place in 2003. For our task, such specific event extractions which are associated with one certain timestamp are desirable. For the sake of distinguishing such extractions from the generic ones, a MaxEnt classifier is used to classify extractions as either specific ones or generic ones.

Training Set Generation A training set is indispensable for training a MaxEnt classifier. In order to generate training examples, we performed HAC discussed in Section 3.1.2 for event coreference resolution on extractions from all documents written in May and June of 1995-1999 and then analyzed each cluster. If extractions in a cluster have different timestamps, then the extractions in this cluster will be labeled as generic extractions (negative); otherwise, extractions in the cluster are labeled as specific ones (positive). In this way, the training set can be generated without manually labeling. To avoid bias of positive and negative examples, we sampled 3,500 positive examples and 3,500 negative examples to train the model.

Feature Selection The following features were selected for training:

Named Entities: People and places are often discussed during specific time periods, particularly in news genre. Intuitively, if an extraction contains specific named entities then this extraction is less likely to be a generic event. If an extraction contains named entities, types and uninterrupted tokens of the named entities will be included as features.

Numeral: According to our analysis of the training set generated by the above-mentioned method, generic extractions usually contain numerals. For example, the extraction "15 people died in this accident" and the extraction "225 people died in this accident" have the same tokens except numerals and they are labeled as a generic event because they are clustered into one group due to high similarity but they in fact refer to different events happening in different years. Therefore, if an extraction contains numerals, the feature "NUM" will be included.

Bag of words: Bag of words can also be an indicator of specific extractions and generic ones. For example, an extraction containing 'stock', 'index', 'fell' and 'exchange' is probably a generic one.

The model obtained after training can be used to predict whether an extraction is a specific one. We define $P(S = 1|ex)$ as the probability that an extraction is a specific one, which can be provided by the classifier. Extractions whose probability to be a specific one is less than 0.05 are filtered out. For the other extractions, this probability is used as a part of the decay factor in the confidence boosting model, which will be discussed in detail in Section 3.2.

3.2 Confidence boosting

After extracting and processing the event extractions, relative temporal relations between documents and events can be constructed. This can be formally represented by a bipartite graph G=$\langle V, E\rangle$. There are two kinds of nodes on the bipartite graph: document nodes and event nodes. Slightly different with the event node mentioned in Section 2, an event node in practice is a cluster of coreferential extractions and it can be connected with multiple document nodes. Note that the bipartite graph does not contain any isolate node. For briefness, we define $DNode$ as the set of document nodes and $ENode$ as the set of event nodes. The set of edges E is formally defined as follows:

$$E = \{e_{ij}, e_{ji} | i \in DNode, j \in ENode, r_{i,j} \in R\}$$

where R is the set of relative temporal relations defined as Section 3.1.1.

3.2.1 Confidence and decay factor

As mentioned in Section 2, the confidence of a node measures the credibility of the predicted timestamp. According to the definition, we set the confidence of initially labeled nodes to 1 and set confidence of nodes without any timestamp to 0 in practice. When the timestamp of a node is propagated to another node, its confidence will be propagated to the target node with some loss, as discussed in Section 2. The confidence loss is caused by two factors in practice. The first one is the credibility of the relative temporal relation between two nodes and the other one depends on whether an extraction refers to a specific event.

Relative temporal relations between documents and extractions we mined using the rule-based method in Section 3.1.1 are not absolutely correct. The credibility of the relations has an effect on the confidence decay. Formally, we used $\pi(i, j)$ to denote the credibility of the relative temporal relation between node i and node j. The credibility of a relative temporal relation in each case can be estimated through table 2. If the credibility of the relative temporal relation between i and j is low, propagation from node i to j probably leads to error. Therefore, the confidence loss should be much in this case. On contrary, if the relation is highly credible, it will be less likely that propagation errors occur. Therefore, the confidence loss should be little.

In addition, whether an extraction refers to a generic event or a specific one exerts an impact on the confidence loss. If an extraction refers to a generic event, then the extractions in the same cluster with it probably have different timestamps. Since our propagation model assumes that extractions in a cluster are coreferent and thus they should have the same timestamp, propagations from a generic event node are very likely to result in propagation errors. Therefore, the timestamp of a generic event node in fact is less credible for propagations and confidence of such event nodes should be low for limiting propagations from the nodes. For this reason, propagation from a document node to a generic event node leads to much loss of confidence. We define the probability that an event node refers to a specific event as follows:

$$P(S = 1|enode) = \frac{1}{|\mathbb{C}|} \sum_{ex \in \mathbb{C}} P(S = 1|ex)$$

where $\mathbb{C}$ is the set of extractions in the event node and $P(S = 1|ex)$ is the probability that an extraction refers to a specific event, which can be provided by the MaxEnt classifier discussed in Section 3.1.3.

Considering the two factors for confidence loss, we formally define the decay factor by (1).

$$\sigma(s, t) = \begin{cases} \pi(s, t) & \textbf{if } t \in DNode \\ \pi(s, t) \times P(S = 1|t) & \textbf{otherwise} \end{cases} \tag{1}$$

3.2.2 Confidence boosting algorithm

In confidence boosting model, the propagation from i to j will occur only if the CB condition is satisfied. The confidence boosting propagation process can be described as figure 4.

```
Algorithm: Confidence Boosting
Input: Array $Y$, Array $c$, Array $\delta$, Array $\sigma$
Output: Array $Y$
    Initialize Array $c$ and Array $Y$
    while $\exists i, j$ s.t. CB condition
        $Y(j) = Y(i) + \delta(i, j)$
        $c(j) = c(i) \times \sigma(i, j)$
    end while
```

Figure 4: Algorithm of confidence boosting

Whenever timestamps are propagated to other nodes, the global confidence of the bipartite graph will increase. For this reason, this propagation process is called confidence boosting. In this model, a node with high confidence is more active than nodes with low confidence to propagate its timestamp. Moreover, a prediction with low confidence can be corrected by the prediction with high confidence. Therefore, the confidence boosting model can alleviate the problem of propagation of errors to some extent and handle conflict of predictions. Thus, it can propagate timestamps more credibly than the BFS-based model. It can also be proved that each node on the bipartite graph must reach the highest confidence it can reach so that the global confidence of the bipartite graph must be optimal when the confidence boosting propagation process ends regardless of propagation orders, which will be discussed in Section 3.2.3.

3.2.3 Proof of the optimality of confidence boosting

Proof by contradiction can be used to prove that propagation orders do not affect the optimality of the confidence boosting model.

Proof Assume by contradiction that there is some node that does not reach its highest confidence it can reach when a confidence boosting process in propagation order A ends:

$$\exists v_t \text{ s.t. } c_A(v_t) < c^*(v_t)$$

where $c_A(v_t)$ is the confidence of v_t when the propagation process in order A ends and $c^*(v_t)$ is the highest confidence that v_t can reach. Assume that $(v_1, v_2, \cdots, v_{t-1}, v_t)$ is the optimal propagation path from the propagation source node v_1 to the node v_t that leads to the highest confidence of v_t, which means that $c^*(v_t) = c^*(v_{t-1}) \times \sigma(v_{t-1}, v_t)$, $c^*(v_{t-1}) = c^*(v_{t-2}) \times \sigma(v_{t-2}, v_{t-1})$, ..., $c^*(v_2) = c^*(v_1) \times \sigma(v_1, v_2)$. Then according to CB condition, since $c_A(v_{t-1}) \times \sigma(v_{t-1}, v_t) \leq c_A(v_t) < c^*(v_t) = c^*(v_{t-1}) \times \sigma(v_{t-1}, v_t)$, the inequality $c_A(v_{t-1}) < c^*(v_{t-1})$ must hold. Similarly, it can be easily inferred that $c_A(v_{t-2}) < c^*(v_{t-2})$ and finally $c_A(v_1) < c^*(v_1)$. Since v_1 is the source node whose timestamp is initially labeled and its confidence is 1, the inequality $c_A(v_1) < c^*(v_1)$ cannot hold. Thus, the assumption that $c_A(v_t) < c^*(v_t)$ cannot be satisfied. Therefore, it can be proved that each node on the bipartite graph must reach the highest confidence it can reach so that the global confidence of the bipartite graph must be optimal when confidence boosting propagation process ends no matter what order time labels are propagated in.

4 Experiments

In this section, we evaluate the performance of our time label propagation models and different automatic document dating models on the Gigaword dataset. We first present the experimental setting. Then we show experimental results and perform an analysis.

4.1 Experimental Setting

Dataset To simulate the environment of the web where data is very redundant, we use all documents written in April, June, July and September of 2000-2004 of Gigaword Corpus as dataset instead of sampling a subset of documents from each period. The dataset contains 900,199 news articles.

Pre-processing Many extractions extracted by ReVerb are short and uninformative and do not carry any valuable information for propagating temporal information. Also, some extractions do not refer to events which already happened. These extractions may affect the performance of event coreference resolution and the rule-based method proposed in Section 3.1.1 for mining relative temporal relations. Therefore, we filter out these undesirable extractions in advance with a rule-based method. The rules are shown in table 3. This preprocessing removes large numbers of "bad" extractions which are undesirable for our task. As a result, not only computation efficiency but also precision of event coreference resolution will be improved.

Rule1	If the number of tokens of the extraction is less than 5 then this extraction will be filtered out.
Rule2	If the maximum idf of terms of the extraction is less than 3.0 then this extraction will be filtered out.
Rule3	If the tense of the extraction is not past tense then this extraction will be filtered out.
Rule4	If the extraction is the content of direct quotation then this extraction will be filtered out.

Table 3: Pre-processing Rules

$\|DNode\|$	550,124
$\|ENode\|$	968,064
$\|E\|$	3,104,666

Table 4: Basic information of the bi-partite graph

Basic information of the document-event bipartite graph constructed is shown in table 4.

Evaluation To evaluate the performance of the propagation models for the task of dating on different sizes of the training set, we used different sizes of the labeled documents for training and considered the remaining documents as the test set. Note that the training set is randomly sampled from the dataset. To be more persuasive, we repeated above experiments for five times.

However, in the time label propagation process, not all documents can be labeled. For those documents which cannot be labeled in the process of propagation, a MaxEnt classifier serves as a complementary approach to predict their timestamps. For the MaxEnt classifier, unigrams and named entities are simply selected as features and the initially labeled documents as well as documents labeled during propagation process are used for training.

Baseline methods are temporal language models proposed by de Jong et al. (2005) and the state-of-the-art discriminative classifier with linguistic features and temporal constraints which was proposed

Initially Labeled	1k	5k	10k	50k	100k	200k	500k
Reached Min	443980	448653	453022	484562	518603	599724	732701
Reached Max	444266	448998	454028	484996	519333	579878	732799
Reached Avg	444107	448742	453786	484622	519110	579835	732758
Prop Ratio	444.1	89.7	45.4	9.7	5.2	2.9	1.5
Prop acc(BFS)	0.438	0.515	0.551	0.646	0.691	0.725	0.775
Prop acc(CB)	0.494	0.569	0.603	0.701	0.746	0.776	0.807

Table 5: Performance of Propagation

Initially Labeled	1k	5k	10k	50k	100k	200k	500k
Temporal LMs	0.277	0.323	0.353	0.412	0.422	0.425	0.420
Maxent(Unigrams)	0.326	0.378	0.407	0.486	0.517	0.553	0.590
Maxent(Unigrams+NER)	0.331	0.383	0.418	0.506	0.549	0.590	0.665
Chambers's	0.331	0.386	0.423	0.524	0.571	0.615	0.690
BFS+Maxent	0.459	0.508	0.533	0.595	0.626	0.658	0.707
CB+Maxent	**0.486**	**0.535**	**0.559**	**0.624**	**0.655**	**0.685**	**0.726**

Table 6: Overall accuracy of dating models

by Nathanael Chambers (2012). In Chambers's joint model, the interpolation parameter λ is set to 0.35 which is considered optimal in his work.

4.2 Experimental Results

Table 5 shows the performance of propagation models where *Reached* denotes the number of documents labeled when the propagation process ends, *prop ratio* and *prop accuracy* are defined as follows:

$$\text{Prop Ratio} = \frac{\#ReachedDocNodes}{\#LabeledDocNodes}$$

$$\text{Prop Accuracy} = \frac{\#CorrectDocNodes - \#LabeledDocNodes}{\#ReachedDocNodes - \#LabeledDocNodes}$$

where $\#LabeledDocNodes$ is the number of initially labeled document nodes which are documents in the training set and $\#ReachedDocNodes$ is the number of document nodes labeled when the propagation process ends.

Note that prop ratio and accuracy in table 5 are the mean of the prop ratio and accuracy of the five groups of experiments. It is clear that confidence boosting model improves the prop accuracy over BFS-based model. When only 1,000 documents are initially labeled with timestamps, the confidence boosting model can propagate their timestamps to more than 400,000 documents with an accuracy of 0.494, approximately 12.8% relative improvement over the BFS counterpart, which proves effectiveness of the confidence boosting model.

However, as shown in table 5, hardly can the propagation process propagate timestamps to all documents. One reason is that the number of document nodes on the bipartite graph is only 550,124, approximately 61.1% of all documents. The other documents may not mention events which are also mentioned by other documents, which means they are isolate and thus are excluded from the bipartite graph. Also, the event coreference resolution phase does not guarantee finding all coreferential extractions; in other words, recall of event coreference resolution is not 100%. The other reason is that some documents are unreachable from the initially labeled nodes even if they are in the bipartite graph.

The overall accuracy of different dating models is shown in table 6. As with table 5, overall accuracy in table 6 is the average performance of models in the five groups of experiments. As reported by Nathanael Chambers (2012), the discriminative classifier performs much better than the temporal language models on the Gigaword dataset. In the case of 500,000 training examples, the Maxent classifier using unigram features outperforms the temporal language models by 40.5% relative accuracy. If the size of the training set is large enough, named entities and linguistic features as well as temporal constraints will improve the overall accuracy significantly. However, if the size of the training set is small, these features will not result in much improvement.

Compared with the previous models, the propagation models predict the document timestamps much more accurately especially in the case where the size of the training set is small. When the size of the training set is 1,000, our BFS-based model and confidence boosting model combined with the MaxEnt classifier outperform Chambers's joint model which is considered the state-of-the-art model for the task of automatic dating of documents by 38.7% and 46.8% relative accuracy respectively. This is because the feature-based methods are not very reliable especially when the size of the training set is small. In contrast, our propagation models can predict timestamps of documents with an understanding of document content, which allows our method to date documents more credibly than the baseline methods. Also, by comparing table 5 with table 6, it can be found that prop accuracy is almost always higher than overall accuracy, which also verifies that the propagation models are more credible for dating document than the feature-based models. Moreover, data is so redundant that a great number of documents can be connected with events they share. Therefore, even if a small number of documents are labeled, the labeled information can be propagated to large numbers of articles through the connections between documents and events according to relative time relations. Even if the size of the training set is large, e.g. 500,000, our propagation models still outperform the state-of-the-art dating method. Additionally, some event nodes on the bipartite graph may be labeled with a timestamp during the process of propagation as a byproduct. The temporal information of the events would be useful for other temporal analysis tasks.

5 Related Work

In addition to work of de Jong et al. (2005) and Chambers (2012) introduced in previous sections, there is also other research focusing on the task of document dating. Kanhabua and Norvag (2009) improved temporal language models by incorporating temporal entropy and search statistics and applying two filtering techniques to the unigrams in the model. Kumar et al. (2011) is also based on the temporal language models, but more historically-oriented, which models the timeline from the present day back to the 18th century. In addition, they used KL-divergence instead of normalized log likelihood ratio to measure differences between a document and a time period's language model.

However, these methods are based on temporal language models so they also suffer from the problem of the method of de Jong et al. (2005). Therefore, they inevitably make wrong predictions in some cases, just as mentioned in Section 1. Compared with these methods, our event-based propagation models exploit relative temporal relations between documents and events for dating document on a basis of an understanding of document content, which is more reasonable and also proved to be more effective by the experimental results.

6 Conclusion

The main contribution of this paper is exploiting relative temporal relations between events and documents for the document dating task. Different with the conventional work which dates documents with feature-based methods, we proposed an event-based time label propagation model called confidence boosting in which timestamps are propagated on a document-event bipartite graph according to relative temporal relations between documents and events for dating documents on a basis of an understanding of document content. We discussed challenges for the propagation models and gave the corresponding solutions in detail. The experimental results show that our event-based propagation model can predict document timestamps in high accuracy and the model combined with a MaxEnt classifier outperforms the state-of-the-art method on a data-redundant dataset.

Acknowledgements

We thank the anonymous reviewers for their valuable suggestions. This paper is supported by NSFC Project 61075067, NSFC Project 61273318 and National Key Technology R&D Program (No: 2011BAH10B04-03).

References

Klaus Berberich, Srikanta Bedathur, Thomas Neumann, and Gerhard Weikum. 2007. A time machine for text search. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 519–526. ACM.

Giang Binh Tran, Mohammad Alrifai, and Dat Quoc Nguyen. 2013. Predicting relevant news events for timeline summaries. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 91–92. International World Wide Web Conferences Steering Committee.

Nathanael Chambers and Dan Jurafsky. 2008. Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 698–706. Association for Computational Linguistics.

Nathanael Chambers. 2012. Labeling documents with timestamps: Learning from their time expressions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 98–106. Association for Computational Linguistics.

FMG de Jong, Henning Rode, and Djoerd Hiemstra. 2005. Temporal language models for the disclosure of historical text. Royal Netherlands Academy of Arts and Sciences.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*.

Po Hu, Minlie Huang, Peng Xu, Weichang Li, Adam K Usadi, and Xiaoyan Zhu. 2011. Generating breakpoint-based timeline overview for news topic retrospection. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 260–269. IEEE.

Pawel Jan Kalczynski and Amy Chou. 2005. Temporal document retrieval model for business news archives. *Information processing management*, 41(3):635–650.

Nattiya Kanhabua and Kjetil Nørvåg. 2009. Using temporal language models for document dating. In *Machine Learning and Knowledge Discovery in Databases*, pages 738–741. Springer.

Abhimanu Kumar, Matthew Lease, and Jason Baldridge. 2011. Supervised language modeling for temporal resolution of texts. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2069–2072. ACM.

Katsumasa Yoshikawa, Sebastian Riedel, Masayuki Asahara, and Yuji Matsumoto. 2009. Jointly identifying temporal relations with markov logic. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language: Volume 1-Volume 1*, pages 405–413. Association for Computational Linguistics.