Omnia Mutantur, Nihil Interit: Connecting Past with Present by Finding Corresponding Terms across Time

Yating Zhang[*], Adam Jatowt[*], Sourav S Bhowmick[+], Katsumi Tanaka[*]
[*]School of Informatics, Kyoto University
[+]School of Computer Engineering, Nanyang Technological University
{zhang,adam,tanaka}@dl.kuis.kyoto-u.ac.jp
assourav@ntu.edu.sg

Abstract

In the current fast-paced world, people tend to possess limited knowledge about things from the past. For example, some young users may not know that Walkman played similar function as iPod does nowadays. In this paper, we approach the *temporal correspondence problem* in which, given an input term (e.g., *iPod*) and the target time (e.g. 1980s), the task is to find the counterpart of the query that existed in the target time. We propose an approach that transforms word contexts across time based on their neural network representations. We then experimentally demonstrate the effectiveness of our method on the New York Times Annotated Corpus.

1 Introduction

What music device 30 years ago played similar role as iPod does nowadays? Who are today's Beatles? Who was a counterpart of President Chirac in 1988? These and many other similar questions may be difficult to answer by average users (especially, by young ones). This is because people tend to possess less knowledge about the past than about the contemporary time.

In this work we propose an effective method to solve the problem of finding counterpart terms across time. In particular, for an input pair of a term (e.g., *iPod*) and the target time (e.g. 1980s), we find the corresponding term that existed in the target time (*walkman*). We consider temporal counterparts to be terms which are semantically similar, yet, which existed in different time.

Knowledge of temporal counterparts can help to alleviate the problem of terminology gap for users searching within temporal document collections such as archives. For example, given a user's query and the target time frame, a new modified query that represents the same meaning could be suggested to improve search results. Essentially, it would mean letting searchers use the knowledge they possess on the current world to perform search within unknown collections such as ones containing documents from the distant past. Furthermore, solving temporal correspondence problem can help timeline construction, temporal summarization, reference forecasting and can have applications in education.

The problem of temporal counterpart detection is however not trivial. The key difficulty comes from the change of the entire context that results in low overlap of context across time. In other words, it is difficult to find temporal counterpart terms by directly comparing context vectors across time. This fact is nicely portrayed by the Latin proverb: "omnia mutantur, nihil interit" (in English: "everything changes, nothing perishes") which indicates that there are no completely static things, yet, many things and concepts are still similar across time. Another challenge is the lack of training data. If we have had enough training pairs of input terms and their temporal counterparts, then it would have become possible to represent the task as a typical machine learning problem. However, it is difficult to collect multiple training pairs over various domains and for arbitrary time.

In view of the challenges mentioned above, we propose an approach that transforms term representations from one vector space (e.g., one derived from the present documents) to another vector space (e.g., one obtained from the past documents). Terms in both the vector spaces are represented by the distributed vector representation (Mikolov et al. 2013a; Mikolov et al. 2013c). Our method then matches the terms by comparing their relative positions in the vector spaces of different time periods alleviating the problem of low overlap between word contexts over time. It also does not require to manually prepare seed pairs of temporal counterparts. We further improve this method by automatically generating reference points that more precisely represent target terms in the form of local graphs. In result, our approach consists of finding *global* and *local* correspondence between terms over time.

To sum up, we make the following contributions in this paper: (1) we propose an efficient method to find temporal counterparts by transforming the representation of terms within different temporal spaces, (2) we then enhance the *global correspondence* method by considering also the local context of terms (*local correspondence*) and (3) we perform extensive experiments on the New York Times Annotated Corpus (Sandhaus, 2008), including the search from the present to the past and vice versa, which prove the effectiveness of our approach.

2 Global Correspondence Across Time

Let the *base time* denoted as T^B mean the time period associated with the input term and let the *target time*, T^T, mean the time period in which we want to find this term's counterparts. Typically, for users, the base time is the present time and the target time is some selected time period in the past. Note however, that we do not impose any restriction on the order and the distance of the both times. Hence, it is possible to search for present counterparts of terms that existed in the past.

In our approach we first represent all the terms in the base time and in the target time within their respective semantic vector spaces, χ^B and χ^T. Then, we construct a transformation matrix to bridge the two vector spaces. Algorithm 1 summarizes the procedures needed to compute the global transformation. We will explain it in Section 2.1 and 2.2.

Algorithm 1 Overview of Global Transformation

Input: query q, base time T^B and target time T^T

1. Construct word representation model for corpus in the base time, $D(T^B)$, and in the target time, $D(T^T)$. (**Section 2.1**)
2. Construct transformation matrix $\boldsymbol{M}$ between $D(T^B)$ and $D(T^T)$ by first collecting ***CFTs*** as training pairs and then learning $\boldsymbol{M}$ using **Eq. 1**. (**Section 2.2**)
3. Rank the words in target time by their correspondence scores (**Eq. 2**)

Output: ranked list of temporal counterparts

2.1 Vector space word representations

Distributed representation of words by neural network was first proposed by Rumelhart et al. (1986). More recently, Mikolov et al. (2013a, 2013c) introduced the *Skip-gram model* which utilizes a simplified neural network architecture for learning vector representations of words from unstructured text data. We apply this model due to its advantages: (1) it can capture precise semantic word relationships; (2) due to the simplified neural network architecture, the model can easily scale to millions of words. After applying the Skip-gram model, the documents in the base time, $D(T^B)$, are converted to a $m \times p$ matrix where n is the vocabulary size and p are the dimensions of feature vectors. Similarly, the documents in the target time, $D(T^T)$, are represented as a $n \times q$ matrix (as shown in Fig. 1).

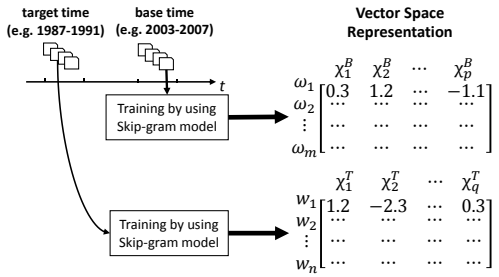

Figure 1: Word vector representations for the base and the target time.

2.2 Transformation across vector spaces

Our goal is to compare words in the base time and the target time in order to find temporal counterparts. However, it is impossible to directly compare words in two different semantic vector spaces, as the features in both spaces have no direct correspondence between each other (as can be seen in Fig. 1). To solve this problem, we propose to train a transformation matrix in order to build the connection between different vector spaces. The key idea is that the relative positions of words in each vector space should remain more or less stable. In other words, a temporal counterpart term should have similar relative position in its own vector space as the position of the queried term in the base time space. Fig. 2 conceptually portrays this idea as the correspondence between the context of *Walkman* and the context of *iPod* (only two dimensions are shown for simplicity).

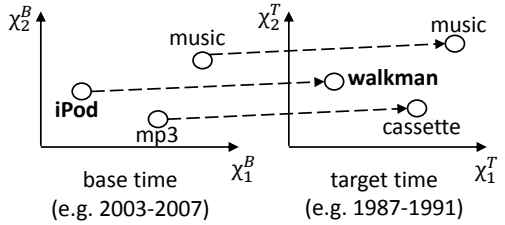

Figure 2: Conceptual view of the across-time transformation by matching similar relative geometric positions in each space.

Our task is then to train the transformation matrix to automatically "rotate" the base vector space

into the target vector space. Suppose we have K pairs of temporal counterparts $\{(\omega_1, w_1),\ldots,(\omega_k, w_k,)\}$ where ω_i is a base time term and w_i is its counterpart in the target time. Then the transformation matrix $\boldsymbol{M}$ can be computed by minimizing the differences between $\boldsymbol{M}\cdot\omega_i$ and w_i as given in Eq. 1. The latter part of Eq. 1 is added as regularization to overcome the problem of overfitting. Intuitively, matrix M is obtained by making sure that the sum of Euclidean 2-norms between transformed query vectors and their counterparts is minimal on K seed query-counterpart pairs. Eq.1 is used for solving regularized least squares problem (γ equals to 0.02).

$$M = \arg\min_{M} \sum_{i=1}^{K} \|M \cdot \omega_i - w_i\|_2^2 + \gamma \|M\|_2^2 \qquad (1)$$

However, as mentioned before, the other challenge is that the training pairs are difficult to be obtained. It is non-trivial to prepare large enough training data that would also cover various domains and any possible combinations of the base and target time periods. We apply here a simple trick that performs reasonably well. We select terms that (a) have the same syntactic forms in the base and the target time periods and (b) are frequent in the both time periods. Such *Common Frequent Terms (CFT*s*)* are then used as the training data. Essentially, we assume here that very frequent terms (e.g., *man*, *women*, *water*, *dog*, *see*, *three*) change their meanings only to small extent. The reasoning is that the more frequently the word is used, the harder is to change its dominant meaning (or the longer time it takes to make the meaning shift) as the word is commonly used by many people. The phenomenon that words used more often in everyday language had evolved more slowly has been observed in several languages including English, Spanish, Russian and Greek (Pargel et al., 2007; Lieberman et al. 2007). Then, using the common frequent terms as the training pairs, we solve Eq. 1 as the least squares problem. Note that the number of *CFT*s is heuristically decided. In Sec. 5 we discuss transformation performance with regards to different numbers of *CFT*s.

After obtaining matrix $\boldsymbol{M}$, we can then transform the base time term, q, first by multiplying its vector representation with the transformation matrix $\boldsymbol{M}$, and then by calculating the cosine similarity between such transformed vector and the vectors of all the terms in the target time. We call the result of this similarity comparison the *correspondence score* between the input term q in the base time and a given term w in the target time (see Eq. 2). A term which has the highest correspondence score could be then considered as temporal counterpart of q.

$$Correspondence(q, w) = \cos(M \cdot q, w) \qquad (2)$$

3 Local Correspondence across Time

The method described above computes "global similarity" between terms across time. In result, the discovered counterparts can be similar to the query term for variety of reasons, some of which may not always lead to the best results. For instance, the global transformation finds *VCR* as the temporal counterpart of *iPod* in 1980s simply because both of them can have recording and playback functions. *Macintosh* is another term judged to be strongly corresponding to *iPod* since both are produced by *Apple*. Clearly, although *VCR* and *Macintosh* are somewhat similar to *iPod,* they are far from being its counterparts. The global transformation, as presented in the previous section, may thus fail to find correct counterparts due to neglecting fundamental relations between a query term and its context.

Inspired by these observations, we propose another method for leveraging the informative context terms of an input query term called *reference points*. They are used to help mapping the query to its correct temporal counterpart by considering the relation between the query and the reference points. We call this kind of similarity matching as *local correspondence* in contrast to *global correspondence* described in Sec. 2. In the following sub-sections, we first introduce the desired characteristics of the *reference points* and we then propose three computation methods for selecting them. Finally, we describe how to find temporal counterparts using the selected reference points. Algorithm 2 shows the process of computing the local transformation.

Algorithm 2 Overview of Local Transformation

Input: query q, base time T^B and target time T^T

1. Construct the local graph of q by detecting the reference points in the context of q. (**Section 3.1**)
2. Compute similarity of the local graph of q with all the local graphs of candidate temporal counterparts in the target time. (**Section 3.2**)
3. Rank the candidate temporal counterparts in the target time by graph similarity score (**Eq. 4**).

Output: ranked list of temporal counterparts

3.1 Reference points detection

Reference points are terms in the query's context which help to build connection between the query and its temporal counterparts. Reference points should have at least some of the following characteristics: (a) have high relation with the query (b) be sufficiently general and (c) be independent from each other.

Note that it does not mean that the selected reference point should have exactly same surface form across time. Let us consider the previous example query *iPod* and 1980s as the target time. The term *music* could be a candidate reference point for this query. Its temporal counterpart has exactly the same syntax form in the target time (*music*). However, *mp3* could be another reference point. Even though *mp3* did not exist in 1980s, it can still be referred to storage devices at the target time such as *cassette* or *disk* helping thus to find the correct counterparts of *iPod*, that is, *walkman* and *CD player*.

Since different reference points will lead to different answers, we propose three methods for selecting the reference points. Each one considers the previously mentioned characteristics of reference points to different extent. Note that, if necessary, the choice of the references points can be left to users.

Term co-occurrence. The first approach satisfies the reference points' characteristics of being related to the query. To select reference points using this approach we rank context terms by multiplying two factors: *tf(c)* and *relatedness(q,c)*, where *tf(c)* is the frequency of a context term *c*, while *relatedness(q,c)* is the relation strength of *q* and *c* measured by the χ^2 test. The test is conducted based on the hypothesis that $P(c|q)=P(c|\bar{q})$, according to which the term *c* has the same probability of occurring in documents containing query *q* and in the documents not containing *q*. We then use the inverse of the p-value obtained from the test as *relatedness(q,c)*.

Lexico-syntactic patterns. As the second approach we propose using hypernyms of terms. This corresponds to the characteristic of reference points to be general words. General terms are preferred rather than specific or detailed ones since the former are more probable to be associated with correct temporal counterparts[1]. This is because detailed or specific terms are less likely to have corresponding terms in the target time. To detect hypernyms on the fly, we adopt the method proposed by Ohshima et al. (2010) that uses bi-directional lexico-syntactic patterns due to its high speed and the lack of requirements for using external ontologies. The latter is important since, to the best of our knowledge, there are no ready ontology resources for arbitrary periods in the past (e.g., there seems to be no Wordnet for the past).

Semantic clustering. The last method chooses reference points from clusters of context terms. The purpose of applying clustering is to avoid choosing semantically similar reference points. Clustering helps to select typical terms from different sematic clusters to provide diverse informative context.

For grouping the context terms we utilize the bisecting k-means algorithm. It is superior over k-means and the agglomerative approach (Steinbach et al., 2000) in terms of accuracy. The procedure of bisecting k-means is to, first, select a cluster to split and then to utilize the basic k-means to form two sub-clusters. These two steps are repeated until the desired number of clusters is obtained. The distance between any two terms w_1, w_2 is the inverse of cosine similarity between their vector representations.

$$Dist(w_1, w_2) = 1 - \cos(w_1, w_2) \quad (3)$$

3.2 Local graph matching

Formulation. The local graph of query *q* is a star shaped graph, denoted as $S_q^{F_B}$, in which *q* is the internal node, and the set of reference points, $F_B = \{f_1, f_2, \ldots, f_u\}$, are leaf nodes where *u* is the number of reference points. Our objective is to find a local graph $S_w^{F_T}$ in the target vector space that is most similar to $S_q^{F_B}$ in the base vector space. *w* denotes here the temporal counterpart of *q* and F_T is the set of terms in the target vector space that corresponds to F_B.

Algorithm. *Step (1)*: to compare the similarity between two graphs in different vector spaces, every node (i.e. term) in $S_q^{F_B}$ is required to be transformed first to allow for comparison under the same vector space. So the transformed vector representation of *q* becomes $\boldsymbol{M}\cdot q$ and F_B is transformed to $\{\boldsymbol{M}\cdot f_1, \boldsymbol{M}\cdot f_2 \ldots, \boldsymbol{M}\cdot f_u\}$ (recall that $\boldsymbol{M}$ is the transformation matrix). *Step (2)*: for each node in $S_q^{F_B}$, we then choose the top *k* candidate terms with the highest correspondence score in the target space. Note that we would need to perform $k \cdot k^u$

[1] We have experimented with hyponyms and coordinate terms used as reference points and found the results are worse than when using hypernyms.

combinations of nodes (or candidate local graphs) in total, to find the best graph with the highest graph similarity. The computation time becomes then an issue as the number of comparisons grows in polynomial way with the increase in the number of candidate terms. However, we manage to reduce the number of combinations to $k \cdot k \cdot u$ by assuming the reference points be independent of each other. Then, for every selected candidate temporal counterpart, we only choose the set of corresponding terms F_T which maximizes the current graph similarity. By default we set k equal to 1000. The process is shown in Algorithm 3.

Algorithm 3 Local Graph Matching

Input: local graph of q, $S_q^{F_B}$
W = top k corresponding terms of q (by **Eq. 2**)
FF = {top k corresponding terms of each f in reference points F_B={ $f_0, f_1, \ldots, f_u$}} (by **Eq. 2**)
for w = $W[1:k]$ **do**:
  sum_cos = 0 *# total graph similarity score*
  for F = $FF[1:u]$ **do:**
    max_cos = 0 *# current maximum similarity*
    for c = $F[1:k]$ **do:**
      find c which maximizes current graph similarity
    end for
    sum_cos += max_cos
  end for
end for
sort W by sum_cos of each w in W.
Output: sorted W as ranked list of temporal counterparts

Graph similarity computation. To compute the similarity of two star shaped graphs, we take both the *semantic* and *relational similarities* into consideration. Fig. 3 conceptually portrays this idea. Since all the computation is done under the same vector space (after transformation), the semantic meaning is represented by the absolute position of the term, that is, by its vector representation in the vector space. On the other hand, the relation is described by the difference of two term vectors. Finally, the graph similarity function $g(S_q^{F_B}, S_w^{F_T})$ is defined as the combination of the *relational similarity function*, $h(S_q^{F_B}, S_w^{F_T})$, and *semantic similarity function*, $z(S_q^{F_B}, S_w^{F_T})$, as follows:

$$\begin{aligned} g(S_q^{F_B}, S_w^{F_T}) &= (1-\lambda) \cdot h(S_q^{F_B}, S_w^{F_T}) + \lambda \cdot z(S_q^{F_B}, S_w^{F_T}) \\ &= (1-\lambda) \cdot \sum_{f_B \in F_B, f_T \in F_T} \max(R_q^{f_B} \cdot R_w^{f_T}) \\ &+ \lambda \cdot \max(\sum_{f_B \in F_B, f_T \in F_T} \cos(f_B, f_T) + \cos(q, w)) \end{aligned} \quad (4)$$

where $R_q^{f_B}$ is the difference of vectors between q and f_B in F_B represented as [q-f_B]. $R_w^{f_T}$ is the difference of vectors between w and f_T in F_T, [w-f_T], where f_T is selected from k candidates corresponding terms of f_B. f_T maximizes the cosine similarity between [q- f_B] and [w- f_T]. λ is set to 0.5 by default. Intuitively, $S_q^{F_B}$ is a graph composed of query and its reference points, while $S_w^{F_T}$ is a graph containing candidate word w and its reference points. The first maximum in Eq. 4 finds for each reference point in the base time, f_B, the top-k candidate terms corresponding to f_B in the target time. Next, it finds within k such f_T that similarity between [q- f_B] and [w- f_T] is maximum (relational similarity). The second maximum in Eq. 4 is same as the first one with the exception that it computes the semantic similarity instead of the relational similarity. The two summations in Eq. 4 aggregate both the similarity scores over all the reference points.

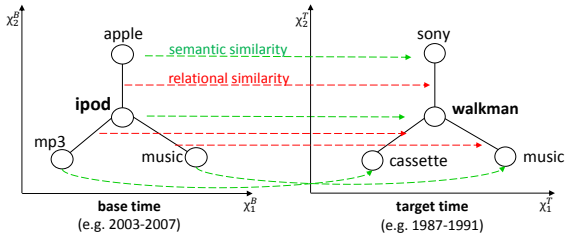

Figure 3: The concept of computing semantic and relational similarity in matching local graphs.

4 Experimental Setup

4.1 Training sets

For the experiments we use the New York Times Annotated Corpus (Sandhaus, 2008). This dataset contains over 1.8 million newspaper articles published between 1987 and 2007. We first divide it into four parts according to article publication time: [1987-1991], [1992-1996], [1997-2001] and [2002-2007]. Each time period contains then around half a million articles. We next train the model of distributed vector representation separately for each time period. The vocabulary size of the entire corpus is 360k, while the vocabulary size of each time period is around 300k.

In the experiments, we first focus on the pair of time periods separated by the longest time gap, that is, [2002, 2007] as the base time and [1987, 1991] as the target time. We also repeat the experiment using more recent target time: [1992, 1996].

q [2002,2007]	*tc* [1987,1991]	BOW (baseline)	LSI-Com (baseline)	LSI-Tran (baseline)	GT (proposed)	LT-Cooc (proposed)	LT-Lex (proposed)	LT-Clust (proposed)
Putin	Yeltsin	1000+	252	353	24	**1**	**1**	**1**
Chirac	Mitterrand	1000+	8	**1**	7	19	**1**	3
iPod	Walkman	1000+	20	131	3	13	**1**	16
Merkel	Kohl	1000+	1000+	537	142	76	**7**	102
Facebook	Usenet	1000+	1000+	1000+	**1**	**1**	**1**	**1**
Linux	Unix	1000+	11	**1**	20	**1**	**1**	**1**
email	letter	1000+	1000+	464	**1**	35	**1**	17
email	mail	1000+	**1**	9	7	**2**	6	11
email	fax	1000+	1000+	10	3	**1**	4	2
Pixar	Tristar	1000+	549	**1**	**1**	**1**	**1**	**1**
Pixar	Disney	1000+	4	4	3	**2**	**2**	4
Serbia	Yugoslavia	1000+	15	1000+	**1**	**1**	**1**	**1**
mp3	compact disk	1000+	56	44	58	**17**	19	22
Rogge	Samaranch	1000+	**4**	22	42	82	34	44
Berlin	Bonn	1000+	43	265	62	**40**	48	56
Czech	Czechoslovakia	1000+	**1**	3	4	**3**	7	4
USB	floppy disk	1000+	209	1000+	20	**1**	**1**	4
spam	junk mail	1000+	1000+	37	5	61	**1**	**1**
Kosovo	Yugoslavia	1000+	59	1000+	14	10	**6**	11

Table 1: Example results where *q* is the input term and *tc* is the matching temporal counterpart. The numbers are the ranks of the correct temporal counterpart in the results ranked by each method. Since we output only the top 1000 results, ranks lower than 1000 are represented as 1000+.

4.2 Test sets

As far as we know there is no standard test bench for temporal correspondence finding. We then had to manually create test sets containing queries in the base time and their correct temporal counterparts in the target time. In this process we used external resources including the Wikipedia, a Web search engine and several historical textbooks. The test terms cover three types of entities: persons, locations and objects.

The examples of the test queries and their temporal counterparts for [1987, 1991] are shown in Table 1 where *q* denotes the input term and *tc* is the correct counterpart. Note that the expected answer is not required to be single neither exhaustive. For example, there can be many answers for the same query term, such as *letter*, *mail*, *fax*, all being commonly used counterparts in 1980s for *email*. Furthermore, as we do not care for recall in this research, we do not require all the correct counterpart terms to be found. In total, there are 95 pairs of terms (query and its counterpart) resulting from 54 input query terms for the task of mapping [2002, 2007] with [1987, 1991], and 50 term pairs created from 25 input query terms for matching [2002, 2007] and [1992, 1996].

4.3 Evaluation measures and baselines

We use the *Mean Reciprocal Rank* (*MRR*) as a main metric to evaluate the ranked search results for each method. MRR is expressed as the mean of the inverse ranks for each test where a correct result appears. It is calculated as follows:

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \quad (5)$$

where $rank_i$ is the rank of a correct counterpart at the *i-th* test. N is the number of query-answer pairs. MRR's values range between [0,1]. The higher the value, the more correct the method is. Besides MRR, we also report precision @1, @5, @10 and @20. They are equal to the rates of tests in which the correct counterpart term *tc* was found in the top 1, 5, 10 and 20 results, respectively.

Baselines. We prepare three baselines:

(1) ***Bag of words approach*** (***BOW***) without transformation: this method directly compares the context of the query in the base time with the context of the candidate term in the target time. We use it to examine whether the distributed vector representation and transformation are necessary.

(2) ***Latent Semantic Indexing (LSI)*** **without transformation (LSI-Com)**: we first merge the documents in the base time and the documents in the target time. Then, we train LSI (Deerwester, 1988) on such combined collection to represent each term by the same distribution of detected topics. We next search for the terms that exist in the target period and that are also semantically similar to the queried terms by comparing their vector

representations. The purpose of using LSI-Com is to check the need for the transformation over time.

(3) ***Latent Semantic Indexing*** (***LSI***) **with transformation** (**LSI-Tran**): we train two LSI models separately on the documents in the base time and the documents in the target time. Then we train the *transformation matrix* in the same way as we did for our proposed methods. Lastly, for a given input query, we compare its transformed vector representation with terms in the target time. LSI-Tran is used to investigate if LSI can be an alternative for the vector representation under our transformation scenario.

Proposed Methods. All our methods use the neural network based term representation. The first one is the method without considering the local context graph called **GT** (see Sec. 2). By testing it we want to investigate the necessity of transforming the context of the query in the target time.

We also test the three variants of the proposed approach that applies the local graph (explained in Sec. 3). The first one, **LT-Lex**, constructs the local graph by using the hypernyms of terms. **LT-Cooc** applies term co-occurrence to select the reference points. Finally, **LT-Clust** clusters the context terms by their semantic meanings and selects the most common term from each cluster.

4.4 Parameter settings

We set the parameters as follows:
(1) *num_of_dim*: we experimentally set the number of dimensions of the Skip-gram model and the number of topics of LSI to be 200.
(2) *num_of_CFTs*: we utilize the top 5% (18k words) of Common Frequent Terms to train the transformation matrix. We have tried other numbers but we found 5% to perform best (see Fig. 4).
(3) *u*: the number of reference points (same as the number of semantic clusters) is set to be 5. According to the results, we found that increasing the number of reference points does not always improve the results. The performance depends rather on whether the reference points are general enough, as too detailed ones hurt the results.

5 Experimental Results

First, we look at the results of finding temporal counterparts in [1987, 1991]. The average scores for each method are shown in Table 2. Table 1 shows detailed results for few example queries.

The main finding is that all our methods outperform the baselines when measured by MRR and by the precisions at different ranks. In the following subsections we discuss the results in detail.

5.1 Context change over time

The first observation is that the task is quite difficult as evidenced by extremely poor performance of the bag of words approach (**BOW**). The correct answers in **BOW** approach are usually found at ranks 10k-30k (recall that the vocabulary size is 360k). This suggests little overlap in the contexts of query and counterpart terms. The fact that all our methods outperform the baselines suggests that the across-time transformation is helpful.

5.2 Using local context graph

We can observe from Table 2 that, in general, using the local context graph improves the results. The best performing approach, **LT-Lex**, improves **GT** method, which uses only global similarity matching, by 24% when measured using MRR. It increases the precision at certain levels of top ranks, especially, at the top 1, where it boosts the performance by 44%. **LT-Lex** uses the hypernyms of query as reference points in the local graph. This suggests that using generalized context terms as reference points is most helpful for finding correct temporal counterparts. On the other hand, **LT-Cooc** and **LT-Clust** usually fail to improve **GT**. It may be because the term co-occurrence and semantic clustering approaches detect less general terms that tend to capture too detailed information which is then poorly related to the temporal counterpart. For example, **LT-Cooc** detects *{music, Apple, computer, digital, iTunes}* as the reference points of the query *iPod*. While *music* is shared by *iPod*'s counterpart (*walkman*) and *Apple* can be considered analogical to *Sony*, other terms (i.e., *computer, digital, iTunes*) are rather too specific and unique for *iPod*.

5.3 Using neural network model

When comparing the results of **LSI-Com** and **LSI-Tran** in Table 2, we can see that using the *transformation* does not help LSI to enhance the performance but, on the contrary, it makes the results worse.

Method	MRR	$P_{@}1$	$P_{@}5$	$P_{@}10$	$P_{@}20$
BOW	4.1E-5	0	0	0	0
LSI-Com	0.206	15.8	27.3	29.5	38.6
LSI-Tran	0.112	7.9	13.6	21.6	22.7
GT	0.298	16.8	44.2	56.8	**73.7**
LT-Cooc	0.283	18.8	35.3	50.6	62.4
LT-Lex	**0.369**	**24.2**	**49.5**	**63.2**	71.6
LT-Clust	0.285	14.7	42.1	55.1	65.2

Table 2: Results of searching from present to past (present: 2002-2007; past: 1987-1991).

Yet, as discussed above, applying the *transformation* is good idea in the case of the Neural Network Model. We believe the reason for this is because it is difficult to perform the global transformation between topics underling the dimensions of LSI, in contrast to transforming "semantic dimensions" of Neural Network Model.

5.4 Effect of the number of CFTs

Fig. 4 shows MRR results for different numbers of *Common Frequent Terms* (*CFT*s) when applying **GT** method. Note that the level of 0.10% (the first point) corresponds to using 658 stop words as seed pairs. As mentioned before, 5% of CFTs allows to obtain the best results.

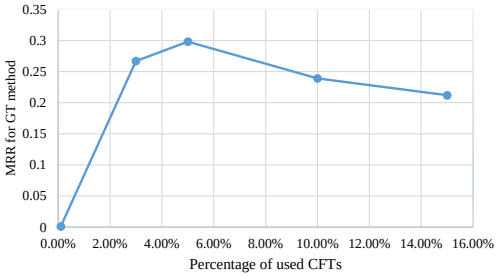

Figure 4: Results of MRR for **GT** method depending on number of used *CFT*s.

5.5 Searching from past to present

We next analyze the case of searching from the past to the present. This scenario may apply to the case of a user (perhaps, an older person) who possesses knowledge about the past term but does not know its modern counterparts.

Table 3 shows the performance. We can see that, again, all our approaches outperform all the baselines using all the measures. **LT-Lex** is the best performing approach, when measured by MRR and P@1 and P@20. **LT-Cooc** this time returns the best results at P@5 and P@10.

Method	MRR	$P_{@}1$	$P_{@}5$	$P_{@}10$	$P_{@}20$
BOW	3.4E-5	0	0	0	0
LSI-Com	0.181	13.2	19.7	28.9	35.5
LSI-Tran	0.109	5.3	17.1	21.1	23.7
GT	0.226	15.2	27.3	33.3	45.5
LT-Cooc	0.231	14.7	**30.7**	**36**	46.7
LT-Lex	**0.235**	**16.7**	28.8	31.8	**48.5**
LT-Clust	0.228	13.6	28.8	31.8	47

Table 3: Average scores of searching from past to present (present: 2002-2007; past: 1987-1991).

The objective of testing the search from the past to present is to prove our methods work in both directions. As for now, we can only conclude the performance is asymmetrical. Yet, we might speculate that, along with the increase in distance, searching from past to present could be harder due to present world becoming relatively more diverse when seen from the distant past.

5.6 Results using different time period

Finally, we perform additional experiment using another target time period [1992, 1996] to verify whether our approach is still superior on different target time. For the experiment we use the best performing baseline listed in Table 2, **LSI-Com**, and the best proposed approach, **LT-Lex**, as well as **GT**. The results are shown in Tables 4 and 5. **LT-Lex** outperforms the other baselines in both the search from the present to the past (Table 4) and from the past to the present (Table 5). Note that since the query-answers pairs for [1992, 1996] are different than ones for [1987, 1991], their results cannot be directly compared.

Method	MRR	$P_{@}1$	$P_{@}5$	$P_{@}10$	$P_{@}20$
LSI-Com	0.115	10.6	14.9	21.3	23.4
GT	0.132	8.5	27.7	40.4	53.2
LT-Lex	**0.169**	**10.6**	**34.1**	**48.9**	**55.3**

Table 4: Results of searching from present to past (present: 2002-2007; past: 1992-1996).

Method	MRR	$P_{@}1$	$P_{@}5$	$P_{@}10$	$P_{@}20$
LSI-Com	0.148	11.6	18.6	23.3	30.2
GT	0.184	11.6	23.3	30.2	44.2
LT-Lex	**0.212**	**14**	**28**	**32.6**	**44.2**

Table 5: Results of searching from past to present (present: 2002-2007; past: 1992-1996).

5.7 Confidence of Results

The approach described in this paper will always try to output some matching terms to a query in the target time period. However in some cases, no term corresponding to the one in the base time existed in the target time (e.g. when the semantic concept behind the term was not yet born or, on the contrary, it has already felt out of use). For example, junk mail may not have any equivalent in texts created around 1800s. A simple solution to this problem would be to use Eqs. 2 and 4 to serve as measures of confidence behind each result in order to decide whether the found counterparts should or not be shown to users. Note however that the scores returned by Eqs. 2 and 4 need to be first normalized according to the distance between the target time and the base time periods.

6 Related Work

Temporal changes in word meaning have been an important topic of study within historical linguistics (Aitchison, 2001; Campbell 2004; Labov, 2010; Hughes, 1988). Some researchers employed computational methods for analyzing changes in word senses over time (Mihalcea and Nastase, 2012; Kim et al., 2014; Jatowt and Duh, 2014; Kulkarni et al., 2015). For example, Mihalcea and Nastase (2012) classified words to one of three past epochs based on words' contexts. Kim et al. (2014) and Kulkarni et al. (2015) computed the degree of meaning change by applying neural networks for word representation. Jatowt and Duh (2014) used also sentiment analysis and word pair comparison for meaning change estimation. Our objective is different as we search for corresponding terms across time, and, in our case, temporal counterparts can have different syntactic forms.

Some works considered computing term similarity across time (Kalurachchi et al., 2010; Kanhabua et al. 2010; Tahmasebi et al. 2012, Berberich et al. 2009). Kalurachchi et al. (2010) proposed to discover semantically identical temporally altering concepts by applying association rule mining, assuming that the concepts referred by similar events (verbs) are semantically related. Kanhabua et al. (2010) discovered the change of terms through the comparison of temporal Wikipedia snapshots. Berberich et al. (2009) approached the problem by introducing a HMM model and measuring the across-time sematic similarity between two terms by comparing the contexts captured by co-occurrence measures. Tahmasebi et al. (2012) improved their approach by first detecting the periods of name change and then by analyzing the contexts during the change periods to find the temporal co-references of different names. There are important differences between those works and ours. First, the previous works mainly focused on detecting changes of the names of the same, single entity over time. For example, the objective was to look for the previous name of Pope Benedict (i.e. Joseph Ratzinger) or the previous name of St. Petersburg (i.e. Leningrad). Second, these approaches relied on applying the co-occurrence statistics according to the intuition that if two terms share similar contexts, then these terms are semantically similar. In our work, we do not require the context to be literally same but to have the same meaning.

Transfer Learning (Pan et al., 2010) is related to some extent to our work. It has been mainly used in tasks such as POS tagging (Blitzer et al., 2006), text classification (Blitzer et al., 2007; Ling et al., 2008; Wang et al., 2011; Xue et al., 2008), learning to rank (Cai et al., 2011; Gao et al., 2010; Wang et al., 2009) and content-based retrieval (Kato et al., 2012). The temporal correspondence problem can be also understood as a transfer learning as it is a search process that uses samples in the base time for inferring correspondent instances existing in the target time. However, the difference is that we do not only consider the structural correspondence but we also utilize the semantic similarity across time.

The idea of distance-preserving projections is also used in automatic translation (Mikolov et al., 2013b). Our research problem is however more difficult and is still unexplored. In the traditional language translation, languages usually share same concepts, while in the across-time translation concepts evolve and thus may be similar but not always same. Furthermore, the lack of training data is another key problem.

7 Conclusions and Future Work

This work approaches the problem of finding temporal counterparts as a way to build a "bridge" across different times. Knowing corresponding terms across time can have direct usage in supporting search within longitudinal document collections or be helpful for constructing evolution timelines. We first discuss the key challenge of the temporal counterpart detection – the fact that contexts of terms change, too. We then propose the global correspondence method using transformation between two vector spaces. Based on this, we then introduce more refined approach of computing the local correspondence. Through experiments we demonstrate that the local correspondence using hypernyms outperforms both the baselines and the global correspondence approach.

In the future, we plan to test our approaches over longer time spans and to design the way to automatically "explain" temporal counterparts by outputting "evidence" terms for clarifying the similarity between the counterparts.

Acknowledgments

We thank Makoto P. Kato for valuable comments. This work was supported in part by Grants-in-Aid for Scientific Research (Nos. 15H01718, 15K12158) from MEXT of Japan and by the JST Research Promotion Program Sakigake: "Analyzing Collective Memory and Developing Methods for Knowledge Extraction from Historical Documents".

References

J. Aitchison, Language Change, Progress or Decay? Cambridge University Press, 2001.

K. Berberich, S. J. Bedathur, M. Sozio and G. Weikum, Bridging the Terminology Gap in Web Archive Search, In *Proc. of WebDB'09*, 2009.

J. Blitzer, M. Dredze, and F. Pereira. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaption for Sentiment Classification. In *Proc. of ACL*, pages 440-447, 2007.

J. Blitzer, R. McDonald, and F. Pereira. Domain adaption with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing. Association for Computational Linguistics (EMNLP)*, pages 120-128, 2006.

P. Cai, W. Gao, A. Zhou et al. Relevant knowledge helps in choosing right teacher: active query selection for ranking adaptation. *In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 115-124, 2001.

L. Campbell, Historical Linguistics, 2nd edition, MIT Press, 2004.

S. Deerwester et al., Improving Information Retrieval with Latent Semantic Indexing, In *Proceedings of the 51st Annual Meeting of the American Society for Information Science*, 25, pages 36–40, 1988.

W. Gao, P. Cai, K.F. Wong et al. Learning to rank only using training data from related domain. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 162-169, 2010.

G. Hughes, Words in Time: A Social History of the English Vocabulary. Basil Blackwell, 1988.

A. Jatowt and K. Duh. A framework for analyzing semantic change of words across time. In *Proc. of JCDL*, pages 229-238, 2014.

A. Kalurachchi, A. S. Varde, S. Bedathur, G. Weikum, J. Peng and A. Feldman, Incorporating Terminology Evolution for Query Translation in Text Retrieval with Association Rules, In *Proceedings of the 19th ACM international Conference on Information and Knowledge Management (CIKM)*, pages 1789-1792, 2010.

N. Kanhabua, K. Nørvåg, Exploiting Time-based Synonyms in Searching Document Archives, In *Proceedings of the 10th annual joint conference on Digital libraries (JCDL)*, pages 79-88, 2010.

M. P. Kato, H. Ohshima and K. Tanaka. Content-based Retrieval for Heterogeneous Domains: Domain Adaption by Relative Aggregation Points. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 811-820, 2012.

Y. Kim, Y-I. Chiu, K. Hanaki, D. Hegde and S. Petrov. Temporal Analysis of Language through Neural Language Models. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pp. 61-65, 2014.

V. Kulkarni, R. Al-Rfou, B. Perozzi, and S. Skiena. 2014. Statistically Significant Detection of Linguistic Change. In *Proc. of WWW*, pages 625-635, 2015.

W. Labov. Principles of Linguistic Change (Social Factors), Wiley-Blackwell, 2010.

E. Lieberman, J.-B. Michel, J. Jackson, T. Tang, M. A. Nowak. Quantifying the evolutionary dynamics of language. *Nature*, 449, 713-716, 2007.

X. Ling, W. Dai, G. R. Xue, Q. Yang and Y. Yu. Spectral domain-transfer learning. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining,* pages 488-496, 2008.

R. Mihalcea, and V. Nastase, "Word Epoch Disambiguation: Finding How Words Change Over Time" in Proceedings of ACL (2) 2012, pp. 259-263, 2012.

T. Mikolov, K. Chen, G. Corrado and J. Dean. Efficient Estimation of Word Representations in Vector Space. In *ICLR Workshop*, 2013a.

T. Mikolov, QV. Le, I. Sutskever. Exploiting similarities among languages for machine translation. CoRR, abs/1309.4168, 2013b.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representation of Phrases and Their Compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3111-3119, 2013c.

H. Ohshima and K. Tanaka. High-speed Detection of Ontological Knowledge and Bi-directional Lexico-Syntactic Patterns from the Web. *Journal of Software*, 5(2): 195-205, 2010.

S. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10): 1345-1359, 2010.

M. Pargel, Q. D. Atkinson and A. Meade. Frequency of word-use predicts rates of lexical evolution

throughout Indo-European history. *Nature*, 449, 717-720, 2007.

D. E. Rumelhart, G. E. Hinton, R.J. Williams. Learning internal representations by error propagation. California Univ, San Diego La Jolla Inst. For Cognitive Science, 1985.

E. Sandhaus. The New York Times Annotated Corpus Overview. The New York Times Company, Research and Development, pp. 1-22, 2008. https://catalog.ldc.upenn.edu/docs/LDC2008T19/new_york_times_annotated_corpus.pdf

M. Steinbach, G. Karypis, V. Kumar. A comparison of document clustering techniques. In *Proc. of KDD workshop on text mining*. 2000, 400(1): 525-526.

N. Tahmasebi, G. Gossen, N. Kanhabua, H. Holzmann, and T. Risse. NEER: An Unsupervised Method for Named Entity Evolution Recognition, In *Proc. of Coling*, pages 2553-2568, 2012.

H. Wang, H. Huang, F. Nie, and C. Ding. Cross-language web page classification via dual knowledge transfer using nonnegative matrix tri-factorization. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 933-942, 2011.

B. Wang, J. Tang, W. Fan, S. Chen, Z. Yang and Y. Liu. Heterogeneous cross domain ranking in latent space. In *Proceedings of the 18th ACM conference on Information and knowledge management (CIKM)*, pages 987-996, 2009.

G. Xue, W. Dai, Q. Yang, and Y. Yu. Topic-bridged plsa for cross-domain text classification. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 627-634, 2008.