Temporal Analog Retrieval using Transformation over Dual Hierarchical Structures

Yating Zhang
RIKEN AIP Center/NAIST, Japan
yating.zhang@riken.jp

Adam Jatowt
Graduate School of Informatics,
Kyoto University, Japan
adam@dl.kuis.kyoto-u.ac.jp

Katsumi Tanaka
Graduate School of Informatics,
Kyoto University, Japan
tanaka.katsumi.85e@st.kyoto-u.ac.jp

ABSTRACT

In recent years, we have witnessed a rapid increase of text content stored in digital archives such as newspaper archives or web archives. Many old documents have been converted to digital form and made accessible online. Due to the passage of time, it is however difficult to effectively perform search within such collections. Users, especially younger ones, may have problems in finding appropriate keywords to perform effective search due to the terminology gap arising between their knowledge and the unfamiliar domain of archival collections. In this paper, we provide a general framework to bridge different domains across-time and, by this, to facilitate search and comparison as if carried in user's familiar domain (i.e., the present). In particular, we propose to find analogical terms across temporal text collections by applying a series of transformation procedures. We develop a cluster-biased transformation technique which makes use of hierarchical cluster structures built on the temporally distributed document collections. Our methods do not need any specially prepared training data and can be applied to diverse collections and time periods. We test the performance of the proposed approaches on the collections separated by both short (e.g., 20 years) and long time gaps (70 years), and we report improvements in range of 18%-27% over short and 56%-92% over long periods when compared to state-of-the-art baselines.

CCS CONCEPTS

•Information systems → Query suggestion; Query reformulation; Similarity measures; Novelty in information retrieval; Question answering;

KEYWORDS

temporal analog, dual hierarchical structure, cluster-biased, heterogeneous document collections

1 INTRODUCTION

A rapid increase of text content stored in digital archives is the result of widespread digitization and content curation initiatives aiming at facilitating access to past documents. Millions of newspapers, books, past snapshots of web pages or other document genres are made accessible and searchable. Unfortunately, most of the current users are not professionals and are not very familiar with the contexts of the past times. When performing search in unfamiliar domains such as document collections spanning several decades users have problems with finding or recalling correct keywords to search with.

To bridge the gap between a user's familiar domain and the unfamiliar search domain, we propose a transformation mechanism based on dual hierarchical structures which automatically transforms terms from one domain to another one for establishing across-domain similarity measure. By this we aim to allow performing search by analogical objects or to help users better understand entities in the past by casting them into the current context. Once the transformation is established, different retrieval requests/applications can be accessed/applied to. For instance, users could issue a query like "`iPod` in 1980s" and the system would suggest similar concept in 1980s such as "`Walkman`". Alternatively, one could enter entity from the past such as "`Samaranch`" (International Olympics Committee president in 1980s-90s) to learn about his present counterpart "`Thomas Bach`". Note that the transformation mechanism does not only help to solve the cold-start problem in search across temporal domains by formulating queries in the form of analogy, but, if successful, it should also help with term understanding and text comparison (e.g., word, sentence, document comparison).

Transformation of text in heterogeneous domains is however not trivial due to strong effect of time elapse, culture differences, etc., suggesting that the direct context comparison will not work. To solve this issue, we apply distributed word embedding technique [28, 29] in order to first set the vocabularies of each domain into their own semantic spaces separately.

To design the mapping function as mentioned above, we first utilize automatically derived training anchor term pairs to construct the transformation matrix for aligning two vector spaces. However the shortcoming of that approach is that it assumes that entire vocabulary in one space follow the same rule (mapping function) in transformation (i.e., a single transformation matrix is used). We thus introduce an advanced approach called *dual hierarchy based term transformation* to relax the above assumption by training multiple transformation matrices biased on specific semantic clusters. This advanced method aims at providing more precise mapping across different vocabulary sets by considering the specificity of transforming different sets of semantics.

To utilize the trained mapping functions (or transformation matrices) for across-domain search, we introduce several retrieval models and similarity metrics based on the above-mentioned transformation mechanisms.

To sum up, our contributions are as follows:

CIKM'17, November 6–10, 2017, Singapore.
 ISBN 978-1-4503-4918-5/17/11…$15.00
DOI: http://dx.doi.org/10.1145/3132847.3132917

(1) We propose an effective and unsupervised framework to transform text across heterogeneous domains such as different time periods, which provides fundamental technique to conduct similarity comparison across vector spaces. To guarantee precise mapping, we introduce an advanced transformation mechanism based on dual hierarchical structures to train specific mapping function for each semantic cluster.
(2) To test the performance of the proposed transformation techniques, we introduce several retrieval models for across-domain search scenarios. Our approaches are unsupervised and do not require any external ontologies, knowledge bases or lexicons (e.g., no need for a Wordnet like lexicon of any POS tagger for the past).
(3) We evaluate the proposed approaches on the unstructured text in temporal collections separated by varying length time gaps.

The reminder of this paper is structured as follows. We start with the introduction of related work in the next section and we formally describe the research problem in Sec. 3. We then discuss the general term transformation technique in Sec. 4. Sec. 5 explains our approach of term transformation based on dual hierarchical structures. Next, we describe the experimental setup and give the evaluation results in Sec. 6 and 7, respectively. We conclude the paper and outline the future work in the last section.

2 RELATED WORK

2.1 Temporal Information Retrieval

Temporal Information Retrieval has become the subject of multiple studies in recent years [10]. Prior research focused on tasks such as time-aware document ranking [5, 11, 19, 24], temporal organization of search results [2, 3], query understanding [11, 26], future information retrieval [4, 16], analyzing semantic shifts over time [15, 22, 23, 27], explaining past documents [35] and so on. Among the above topics, temporal change of the semantic meaning - an emerging topic of study within historical linguistics [1, 9, 14, 25] is relevant to this work. Several researchers employed computational methods for analyzing changes in word senses over time. Mihalcea *et al.* [27] classified words to one of three past epochs based on word contexts. Kim *et al.* [22] and Kulkarni *et al.* [23] computed the degree of meaning change by applying neural networks for word representation. Our objective is however different as we directly search for corresponding terms across time rather than analyze the scope of semantic change of a given target word.

Topic detection and tracking [6, 38, 41] focused on developing methods for tracking changes in the popularity of topics over time given a text corpus. For example, Blei *et al.* [6] and Wang *et al.* [38] extend latent Dirichlet allocation (LDA) [7] to model topic evolution over time. Blei *et al.* assume that the topics in one year are dependent on the topics in the previous year, while Wang and McCallum assume that each topic has its own distribution over time. The objective of those works is to explore *how the topic evolves over time* by tracking topics over continuous time spans. On the other hand, our research is an information retrieval problem to find *semantically equivalent terms in two distant time periods (temporal analog detection).*

Some works have already approached the problem of matching terms across time [5, 17, 18, 34, 39, 40]. Berberich *et al.* [5] proposed to find similar terms by direct context comparison and by applying Hiden Markov Model based model. Kalurachchi *et al.* [17] tried to discover semantically similar concepts by association rule mining under the assumption that concepts associated with the same verbs tend to be similar. Tahmasebi *et al.* [34] extended the work of Berberich *et al.* [5] by detecting bursty time periods when the same concept or the same entity changed its name and applied a rule-based approach for finding synonymous terms in such periods. Another work relied on analyzing revisions in temporal snapshots of Wikipedia to detect name variants of the same objects [18]. All these approaches assume that the same surface form words retain identical semantics over time (essentially assuming a homogeneous document collection). This assumption is too restrictive over longer time spans due to language change as discussed above and, in general, due to the World's change. Furthermore, [5, 17, 34] rely on another simplifying assumption that the contexts of compared terms (terms from different time periods) have relatively high overlap allowing any meaningful comparison. Finally, the proposal of Kanhabua *et al.* [18] is only applicable to short timespans - the last ten years during which Wikipedia existed.

Zhang *et al.* [39, 40] used neural network based term representations for capturing word semantics in different time periods. They first constructed a mapping function to align two vector spaces using Global Transformation (GT) technique to perform mapping across the entire vocabularies in an off-line manner. Then, upon receiving query, an on-line process called Local Transformation (LT) was conducted to perform a more precise mapping by selecting reference terms for improving the temporal analog retrieval. Note that unlike the local transformation approach, our methods work is in an off-line manner without the need for expensive computation during query time. Naturally, same as in the case of Local Transformation [39, 40], an additional on-line computation process can be incorporated to further re-rank the initial candidate results and by these to obtain better results. Compared to Global Transformation our proposal utilizes automatically constructed dual hierarchical structures to harness semantic structures existing in the distinct vector spaces. As it will be demonstrated in the experiments, our proposal significantly outperforms Global Transformation by a wide margin on all the tested time periods and used datasets.

2.2 Domain Adaptation

Several researchers [8, 20, 21, 30] investigated the domain adaptation task. For example, Blitzer *et al.* [8] proposed a Structural Correspondence Learning (SCL) to find correspondences among features from different domains. This was done by modeling correlations of the features with pivot features. Their method was proved to perform well in a discriminative frameworks such as in the task of PoS-tagging. Similarly, Kato *et al.* [20, 21] utilized Relative Aggregation Points (RAP) such as average price, maximum/minimum cost, restaurant categories etc. in different domains as pivot features to detect restaurants in different cities that correspond to a given query restaurant. Both those proposals work in a discriminative learning manner where a conditional probability of instances in a domain is estimated and classified into a certain class. Hence, they can only be applied for the data where instances are

already classified or their distributions over categories are known in corresponding domains. For unstructured text collections (e.g., news archives, online reviews, books), where the entities are not represented by any fixed attributes, it is necessary to user other information and different techniques. Unlike these works, we propose a general framework that leverages the semantics of terms and their relative positions in semantic spaces constructed over distinct time periods for performing cluster-guided transformation. Our method can be thus applied to cases when a query is an arbitrary term such as object or person name, and data is retrieved from any unstructured datasets.

2.3 Analogical Relation Detection

Lastly, analogical relation detection [12, 36, 37] is to some extent related to our work. Structure Mapping Engine (SME) [12] was the original implementation of a well-known Structure Mapping Theory (SMT) [13] that defines the way in which humans perform analogical inference. Latent Relational Mapping Engine (LRME) [37] extended these ideas by extracting lexical patterns in which words co-occur to measure relational similarity of analogous word pairs. These approaches are however always based on a single dataset. Hence, contextual information specific to a particular time period is lost, as it will be later proved in our experiments; and so, the semantic transformation is necessary.

3 PROBLEM STATEMENT

In this section, we formally define the problem of term similarity measurement across heterogeneous domains such as different time periods.

We set two spaces: a base domain (i.e., domain familiar to a user such as the present decade or a time period spanning few recent years) $\mathbf{S^b} = \{\mathbf{w_1^b}, \mathbf{w_2^b}, ..., \mathbf{w_m^b}\}$ ($\mathbf{w_i^b}$ is the vector representation of a term w_i^b, $w_i^b \in$ Vocabulary of $\mathbf{S^b}$) from which the query is selected, and a target domain (user's search domain, i.e., some period in the past) $\mathbf{S^t} = \{\mathbf{w_1^t}, \mathbf{w_2^t}, ..., \mathbf{w_n^t}\}$ ($\mathbf{w_i^t}$ is the vector representation of term w_i^t, $w_i^t \in$ Vocabulary of $\mathbf{S^t}$) where the answer is to be retrieved from.

Term Transformation is a mapping function $M(*)$ used to align the vocabularies of $\mathbf{S^b}$ and $\mathbf{S^t}$ where each of the vector space contains the vector representation of words.

Temporal Analog, also called here a *temporal counterpart*, is defined as a term w^t (e.g., `Walkman`) which is semantically similar to the queried term w^b (e.g., *iPod* in the scenario of searching across time periods: 1980s vs. 2000s). Note that the literal forms of temporal counterparts can be different from each other as long as their meanings remain similar. Moreover their context terms are not required to be literally same.

4 GLOBAL TRANSFORMATION

We focus in this section on constructing the general mapping function between the base time and the target time. This process is query independent and can be carried offline before a user issues a query. We first introduce the way to represent terms in the base time and terms in the target time within their respective semantic vector spaces, $\mathbf{S^b}$ and $\mathbf{S^t}$. Then, we construct a transformation matrix as a general mapping function to bridge the two vector spaces.

4.1 Word Embedding

For capturing word semantics we utilize word embedding techniques. Distributed representation of words by using neural networks was originally proposed in [32]. Mikolov *et al.* [28, 29] improved such representation by introducing Skip-gram model based on a simplified neural network architecture for constructing vector representations of words from unstructured text. Skip-gram model has two important advantages: (1) it captures precise semantic word relationships and (2) it can easily scale to millions of words.

4.2 Transformation based on Anchor Mapping

Our goal is to compare terms in the base space and terms in the target vector space to estimate their similarity and by this to find temporal counterparts. Since, we cannot directly compare words in the two different semantic vector spaces (the features/dimensions have no direct correspondence due to separate training), we train a transformation matrix for building the basic connection between the vector spaces. For this we use a set of training examples called here anchor terms.

However, manually preparing sufficiently large sets of anchor terms that would cover various topics/domains as well as exist in any possible combinations of the base and target spaces requires much effort and resources. We rely here on an approximation procedure for automatically proposing anchor pairs. Specifically, we select terms that have high frequency (e.g., `man`, `city`, `device`, `water`) in both the base and the target spaces. The intuition behind this idea is that terms that are frequent in both spaces are more likely to have stable meaning and also co-occur with many other terms. This observation has been validated by linguistic studies of several Indo-European languages including English which discovered lower semantic drift of frequently used terms [25, 31]. Even if certain anchor term pairs do not retain the same semantics across-time, especially, when separated by relatively long time periods, still, the results should not deteriorate significantly when using sufficiently high number of anchor term pairs.

Suppose there are u pairs of anchor terms $\{(x_1^b, x_1^t),\ldots,(x_u^b, x_u^t)\}$ where x_i^b (e.g., `man`) is an anchor in one space (e.g., 1980s) and x_i^t is its counterpart term, that is, the term with the same literal form (i.e., `man`) in the other space (e.g., 2000s). Transformation matrix $\mathbf{M}$ is established by minimizing the differences between $\mathbf{Mx_i^b}$ and $\mathbf{x_i^t}$ (see Eq. 1). This is realized by minimizing the sum of Euclidean 2-norms between the transformed query vectors and their counterparts. Eq. 1 is used for solving the regularized least squares problem with regularization component used for preventing overfitting ($\gamma = .02$):

$$\mathbf{M} = \underset{\mathbf{M}}{\operatorname{argmin}} \sum_{i=1}^{u} \left\| \mathbf{Mx_i^b} - \mathbf{x_i^t} \right\|_2^2 + \gamma \left\| \mathbf{M} \right\|_2^2 \quad (1)$$

u denotes here the size of anchor term set which contains the top 5%[1] frequent terms in the intersection of vocabularies of the two corpora.

[1]We use 5% as this rate was experimentally verified to result in the best performance for transforming across domains separated by short time gap.

4.3 Retrieval Model for Global Transformation

After obtaining the transformation matrix **M**, we can compute the similarity of a query q in the base space with any term w in the target space by multiplying the query's vector representation with the transformation matrix **M**, and then by calculating the cosine similarity between the transformed vector and w's vector representation denoted by **w**. We call this approach Global Transformation (**GT**).

$$S_{sim}(q, w) = cos(\mathbf{Mq}, \mathbf{w}) \tag{2}$$

5 TRANSFORMATION BASED ON DUAL HIERARCHICAL STRUCTURES

In Sec. 3, we explained the way to construct a single transformation matrix by assuming that all the vocabularies in one space follow the same mapping function, which is obviously a simplified approach. However, we noticed that performance drops for different types of queries when using Global Transformation (**GT**) (e.g., significantly lower performance for person and location queries, while relatively good performance for objects as illustrated in Tab. 5). It suggests the limitations of **GT** in mapping all the terms in semantic space due to its lack of adaptation to different semantic areas (i.e., query independent transformation strategy of **GT**). To solve this problem, we propose an advanced approach, called *dual hierarchy-biased term transformation*, in which we train multiple transformation matrices biased on semantic clusters to which a given query belongs. The motivation behind this approach lies in the notion that *a single transformation matrix is too coarse and too general to serve well for any possible queries.* We believe that the combination of "local" approaches designed for semantic subspaces should achieve better performance (the conceptual comparison between **GT** and dual hierarchy-based term transformation is visualized in Fig. 1). **GT** is actually equivalent to a special case of the dual hierarchy-based approach. It considers only global semantic correspondence across the two vector spaces, while the dual hierarchy based transformation takes into consideration the semantic correspondence biased on semantic clusters (Sec. 5.2) as well as the structural correspondence dependent on each query (Sec. 5.3 and 5.4). The new approach builds upon the key characteristic of word embedding spaces such that semantically similar words are located close to each other [28, 29]. We propose then a cluster-biased approach in which *each semantic cluster should be subject to its own specific transformation mechanism.* In the following sections, we first introduce the way to construct semantic clusters in either space by hierarchical clustering and we then explicate the procedure for establishing the mapping function across clusters in two domains.

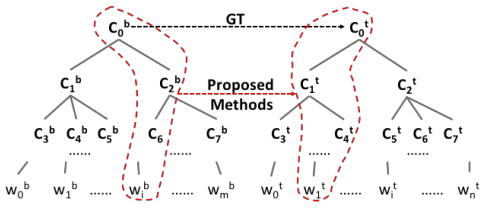

Figure 1: Conceptual comparison between GT and dual hierarchy-based term transformation.

5.1 Hierarchical Clustering in Vector Space

Hierarchical Agglomerative Clustering (HAC), one of the methods of cluster analysis, has been successfully used for building a hierarchy of clusters in a "bottom up" clustering manner. In this paper, we utilize the complete-linkage criterion[2] to determine the distance between clusters when merging the clusters, that is, the minimum distance between elements of each cluster. The distances between words are measured by the inverse of cosine similarity between their word embeddings. HAC processes are carried separately for each vector space. After they are completed, we obtain two hierarchical structures which will support the proposed transformation approach. Each word in either vector space belongs to a hierarchical path of the clusters that spans from each leaf cluster (the word itself) to the root cluster that covers all the words.

5.2 Dual Hierarchy based Term Transformation

As discussed in the beginning of Sec. 4, we are going to establish transformation matrices biased on different semantic clusters. Then the mapping of a given query can be conducted by leveraging the transformation matrices of the semantic clusters where the query belongs to. Different from training of the global transformation which assigns equal importance to all the anchors (see Eq. 1), our idea is to associate weights (λ) to the anchors based on their relation to a given semantic cluster. Each cluster within the hierarchy will then have its own distribution of weights used for biasing the anchors when training the transformation matrix (see Eq. 3).

$$\mathbf{M_k} = \underset{\mathbf{M_k}}{\operatorname{argmin}} \sum_{i=1}^{u} \lambda_{i,k} \left\| \mathbf{M_k x_i^b} - \mathbf{x_i^t} \right\|_2^2 + \gamma \left\| \mathbf{M_k} \right\|_2^2 \tag{3}$$

$\mathbf{M_k}$ is a transformation matrix for cluster C_k^b. $\lambda_{i,k}$ denotes the weight of an anchor x_i biased on cluster C_k^b, which is computed by applying discounting function as shown in Eq. 4. The hypothesis behind the anchor weighting lies in the fact that the anchor which is "closer" to the computed cluster should have higher impact on the transformation of the terms in that cluster. We compute the weight of an anchor for a given cluster by the distance (number of hops) to the nearest cluster containing the anchor within the hierarchy tree. We will illustrate the discounting function by a toy example later. The weights are computed as follows:

$$\lambda_{i,k} = \frac{1}{L_{C_k^b} - L_{C_j^b} + 1} \tag{4}$$

where $L_{C_k^b}$ (or $L_{C_j^b}$) denotes the length (the number of hops) of the shortest path from cluster C_k^b (or C_j^b) to the root of the hierarchy tree; C_j^b represents the cluster (1) which is on the shortest path from C_k^b to the root, (2) which contains the anchor x_i and, at the same time, (3) it is the nearest cluster to C_k^b.

Toy Example. We present a toy example for explaining how to calculate anchor weights for a given cluster. The hierarchy tree of the words in the base space in our example is shown in Fig. 2. We

[2]Complete-linkage method tends to create a fairly balanced tree compared to the single-linkage and is also faster.

are going to compute the anchor weights for the semantic cluster $C_{k=4}$. Suppose the anchors for training the transformation matrix are $\{x_1, x_2, x_3\}$, then Tab. 1 computes the weights of the anchors following Eq. 4. After computing the anchor weights, they are normalized over all the anchors and input to Eq. 3 to obtain the transformation matrix biased on cluster $C_{k=4}$.

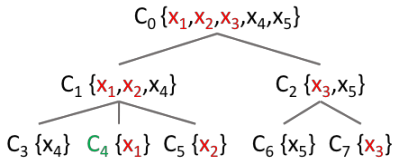

Figure 2: Toy example of hierarchy tree in one space.

Table 1: Toy example of weights calculation for $C_{k=4}$ (for simplicity, we removed the mark b representing the base domain).

anchor	path	C_j	$L_{C_{k=4}} - L_{C_j} + 1$	$\lambda_{i,k=4}$
x_1	$\{C_4, C_1, C_0\}$	C_4	2-2+1=1	1
x_2	$\{C_5, C_1, C_0\}$	C_1	2-1+1=2	1/2
x_3	$\{C_7, C_2, C_0\}$	C_0	2-0+1=3	1/3

Note that when $k = 0$ (root cluster), all anchor weights are equal, which is as the same as the approach of the global transformation introduced in Sec. 3. Global transformation (Eq. 1) can be then regarded as a special case of the dual hierarchy based term transformation.

Based on the obtained transformation matrix $\mathbf{M_k}$ specific for a given semantic cluster C_k^b, the similarity of a query q in the base space with any term w in the target space can be computed in a similar way as the retrieval model for Global Transformation does, but this time utilizing the cluster-biased transformation matrix $\mathbf{M_k}$ (see Eq. 5). We call this approach Hierarchical Term Transformation **HT**.

$$HT(q, w|C_k^b) = cos(\mathbf{M_k q}, \mathbf{w}) \tag{5}$$

Since a term (e.g., query) belongs to many clusters within the hierarchy tree and each cluster has its own transformation matrix, across-time similarity can be computed by applying different combinations of transformation matrices from clusters to which the query belongs. Multiple variants of approaches are possible based on the dual hierarchy framework to generate the ranking list of candidate temporal counterparts. We will discuss several retrieval models in detail in Sec. 5.5.

5.3 Cluster Correspondence

In this section, we propose another signal to improve the computation of the across-time term similarity. We utilize the correspondence between the base space cluster to which the query belongs and the cluster in the target space which contains the given counterpart candidate. We consider that the hierarchy of each vector space can be roughly regarded as a "is-a" relationship tree among clusters where the clusters near the root represent more general concepts. The idea behind the cluster correspondence is as follows: *if there is good correspondence between terms across domains, there should also exist good correspondence in their memberships in the semantic clusters.* Finding cluster mappings should thus help to measure the across-domain similarity from the perspective of structural alignment.

To measure the correspondence between clusters of the both hierarchies, we utilize the membership data of anchors in the clusters. We compute the correspondence by assuming again that anchors remain stable in the semantic hierarchies. A given cluster is represented by a vector of memberships over all the anchors, where 1 or 0 at position i denotes whether the cluster contains the anchor x_i or not. The cluster correspondence is then computed by Eq. 6 where $\mathbf{C_i^b}$ and $\mathbf{C_j^t}$ are vector representations of C_i^b and C_j^t.

$$CC(C_i^b, C_j^t) = cos(\mathbf{C_i^b}, \mathbf{C_j^t}) \tag{6}$$

We will use the cluster correspondence between query and the candidate counterpart in certain variants of the retrieval model.

5.4 Structural Correspondence

Besides Cluster Correspondence **CC** which reflects pairwise semantic similarity of clusters across spaces, another metric called *Structural Correspondence* (**SC**) is set up to measure whether the relative level of the cluster where the query belongs to within its hierarchy is similar to that of its temporal counterpart candidate. We approximate the position of the cluster on the query's path by its relative distance to the root. The position of the cluster on the counterpart's path is measured in a similar way. Then **SC** is computed by Eq. 7 and can be used as the signal for testing structural alignment of clusters when constructing the way for combining results from multiple transformation matrices.

$$SC(C_i^b, C_j^t) = 1 - \left| \frac{L_{C_i^b}}{L_q} - \frac{L_{C_j^t}}{L_w} \right| \tag{7}$$

where C_i^b (C_j^t) is the cluster on query's (or counterpart's) path. $L_{C_i^b}$ ($L_{C_j^t}$) denotes the length (expressed in the number of hops) of the shortest path from cluster C_i^b (C_j^t) to the root in the base space (in the target space). L_q (L_w) represents the length (i.e., hop count) of the shortest path from the leaf node query (counterpart) to the base root (target root).

5.5 Retrieval Model for Dual Hierarchy based Term Transformation

In this section, we discuss the retrieval model by considering the three above-introduced signals to compute across-time term similarity. Eq. 8 computes the term similarity biased on the semantic cluster C_k^b on the path of query q where C_j^t is the cluster on the path of the counterpart w. All the components in Eq. 8 have been previously normalized.

$$\begin{aligned} S_{sim}(q, w|C_k^b) &= HT(q, w|C_k^b) \cdot CC(C_k^b, C_m^t) \cdot SC(C_k^b, C_m^t) \\ \text{where } C_m^t &= \underset{C_m^t \in path(w)}{\text{argmax}} \; CC(C_k^b, C_m^t) \end{aligned} \tag{8}$$

To combine the results from different semantic clusters, we propose to either select the maximum score as the final term similarity degree between q and w, or the summation of the similarity scores

at each hop.

$$S_{sim}(q, w) = \max_{C_k^b \in path(q)} S_{sim}(q, w|C_k^b) \quad (9a)$$

$$S_{sim}(q, w) = \sum_{C_k^b \in path(q)} S_{sim}(q, w|C_k^b) \quad (9b)$$

6 EXPERIMENTS

We conduct experiments on the document collections spanning short time periods as well as on the ones separated by longer time gaps in order to evaluate the performance of the proposed methods.

6.1 Short Time Gap Datasets

We use the New York Times Annotated Corpus [33], which has been frequently utilized in related studies [5, 34, 39]. It contains over 1.8 million newspaper articles published from 1987 to 2007. We test the tasks of searching from [2002,2007] to the two past periods, [1987,1991] and [1992,1996]. Each time period contains around half a million news articles, which is sufficient for training word representations. We train the word embedding model for each time period. On average, time periods have 337k unique terms after removing terms with frequency less than 5. Based on the trained semantic vector spaces, we build hierarchy tree by Hierarchical Agglomerative Clustering (HAC) (complete-linkage) for each space (see Sec. 5.1). The construction of word embedding space as well as the hierarchy trees are conducted offline.

6.2 Long Time Gap Datasets

To experiment with longer time gaps, we use the Times Archive[3], which contains 11 million digitalized news articles published from 1785 to 2009 in the "The Times". We focus on testing the performance of the proposed approaches over the recent 70 years' long period (from 1939 to 2009) during which a large number of news articles have been published and the quality of the archive is still reasonably good[4]. Due to the non-uniform distribution of data over time (recent periods have much larger amount of documents than distant periods, e.g., the sizes of data for [2004,2009] and for [1939,1955] are equal), we decided to split the dataset by size rather than by equal time periods. Considering the total size of the dataset of nearly 400GB, the selection criteria is set to be 20GB. We then take non-overlapping time periods, each containing roughly 20GB of data as time periods. We will experiment on searching from [2004,2009] treated as a base period to the two periods in distant past [1967,1976] and [1939,1955]. The way of training word embedding models and building semantic hierarchy trees for each tested time period are the same as ones for the short time gap datasets.

To alleviate OCR impreciseness, we also propose an unsupervised approach to correct OCR errors by automatically constructing a dictionary for mapping incorrect word spellings into their correct forms. To build the dictionary, we rely on three assumptions: (1) the misspelled terms share similar context with its correctly spelled form. Hence, the incorrectly spelled term should be positioned close to its correct form in the vector space. (2) The misspelled terms have their literal forms similar to the correctly spelled variant (edit distance equals to one in our settings). (3) The correctly spelled term is more dominant (i.e., frequent) compared to its incorrectly spelled versions. In general, we are checking the semantic neighbor set C of any word w[5] and then find such a word in C which meets the above three requirements to consider it as the correct form of w. The constructed mapping dictionary is then applied to the ranked lists of candidate results as post-processing step to replace incorrect spelling results with their correct variants.

6.3 Test Sets

To the best of our knowledge, there are no benchmark datasets for our task. We then construct test sets based on carefully analyzing various sources of history-related knowledge, the Wikipedia as well as using Web search engines. We build test sets separately for each tested period in NYT corpus and in Times Archive. The test sets contain queries in the base time and their temporal counterparts in the target time.

The tests sets used for the experiments on the NYT corpus extend the ones utilized in [39][6], and have in total 225 query-to-answer pairs for [2002,2007]→[1987,1991] and 100 query-to-answer pairs for [2002,2007]→[1992,1996]. Each test set contains three types of entities: *persons*, *locations* and *objects*. Persons include presidents, prime ministers or chancellors of the most developed and populous countries (e.g., `USA`, `UK`, `France`, etc.) as well as the names of popes and FIFA presidents. Locations include names of countries or cities (e.g., `Czechoslovakia`, `Berlin`) that changed their names over time, split into several countries, merged into another political system, or become new capitals. Finally, objects cover instances of devices (e.g., `iPod`, `mobile phone`, `dvd`), concepts (e.g., `email`), companies/institutions (e.g., `NATO`, `Boeing`) or other objects (e.g., `letter`, `euro`). The test pairs are publicly available[7].

Similar to the way discussed above, we also create test sets for the Times Archive. Their size is 108 query-to-answer pairs for the search task: [2004,2009]→[1967,1976] and 109 query-to-answer pairs for [2004,2009]→[1939,1955]. The test pairs are also available online[8].

6.4 Tested Methods

6.4.1 Baselines. We prepare five baselines as follows:

(1) **Word embedding model without transformation (NT)**: **NT** uses distributional representation for capturing word semantics (see Sec. 4.1) same as the proposed methods do. However, instead of training the document collections from two periods separately, it trains a joint vector space by merging the document collections. We can then evaluate the necessity of the transformation by testing this method in comparison to the proposed methods.

(2) **Hidden Markov Model (HMM)** proposed by Berberich *et al.* [5]: the key idea behind this method is to measure the degree of across-time semantic similarity between two terms by comparing their context words based on co-occurrence statistics by utilizing a Hidden Markov Model. We select this approach as a baseline

[3] http://gale.cengage.co.uk/times.aspx/

[4] We have found relatively large number of OCR errors before this date resulting in rapidly deteriorating quality of collection.

[5] C contains the k-Nearest Neighbors of w, where k is set to 5.

[6] Initially, they contained 95 query-to-answer pairs for [2002,2007]→[1987,1991] and 50 query-to-answer pairs for searching from [2002,2007]→[1992,1996].

[7] http://www.dl.kuis.kyoto-u.ac.jp/~adam/temporalsearch_short_extended.txt

[8] http://www.dl.kuis.kyoto-u.ac.jp/~adam/temporalsearch_long_extended.txt

because (a) its objective is identical to ours (searching for temporal counterparts); (b) the input is unstructured data same as one used in our approach (i.e., unprocessed temporal news article archive such as New York Times dataset); (c) this method solves the general problem of temporal counterpart finding, hence, not only finding the name changes of the same entity as in [34].

(3) **Global Transformation** (**GT**) proposed by Zhang *et al.* [39, 40] maps the terms in one vector space to the other by training a global transformation matrix as described in Sec. 4. We set **GT** as a control baseline to verify the effectiveness of our proposed transformation technique based on dual hierarchical structures as described in Sec. 5. We use 5% as the rate of anchor terms for both **GT** and for our proposed methods as this number was found to produce the best results for **GT**.

(4) **Cluster Correspondence** (**CC**) and **Cluster Correspondence with Structural Correspondence** (**CC+SC**) are also tested in order to examine the effect of leveraging only the hierarchy/cluster information without applying semantic transformations.

6.4.2 Proposed Methods. As introduced in Sec. 5, we propose three components to perform across-time transformation: Hierarchical Transformation (**HT**), Cluster Correspondence (**CC**) and Structural Correspondence (**SC**). In the experiments, we test several combinations of these features to evaluate their impact on the performance of search task.

(1) **HT** conducts term transformation at each cluster on the path of query to the root following the base hierarchical tree (see Sec. 5 and Eq. 8 without the **CC** and **SC** components).

(2) **HT+CC** considers the semantic term transformation of clusters in the hierarchy tree that the query belongs to (same as **HT**), however, it also utilizes the cluster correspondence between the cluster the query belongs to and the one that contains the candidate counterpart in target hierarchy tree (see Sec. 5.3 and Eq. 8 without the **SC** component).

(3) **HT+SC** takes into consideration the semantic correspondence and the structural correspondence. It finds counterparts in target hierarchy tree by prioritizing terms that have similar path structure in the target hierarchy as that of query in the base hierarchy tree (see Sec. 5.4 and Eq. 8 without the **CC** component).

(4) **HT+CC+SC** finally combines all the features (see Eq. 8). For each proposed method, the final ranking list is realized either by the maximum score at a certain cluster that the query belongs to (e.g., $\mathbf{HT_{max}}$, $\mathbf{\{HT+CC\}_{max}}$, $\mathbf{\{HT+SC\}_{max}}$, $\mathbf{\{HT+CC+SC\}_{max}}$) or the sum of the score from each hop (cluster) on the query's path (e.g., $\mathbf{HT_{sum}}$, $\mathbf{\{HT+CC\}_{sum}}$, $\mathbf{\{HT+SC\}_{sum}}$, $\mathbf{\{HT+CC+SC\}_{sum}}$) (see Eqs. 9).

7 EVALUATION

7.1 Evaluation Measures

We use Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP) as main measures for evaluating the ranked search results (both range from 0 to 1). The higher the obtained value, the more correct the tested method is. Besides MRR and MAP, we also report precision @1, @5, @10, @20 and @50. The precisions are equal to the rates of queries for which the correct counterpart term is found in the top 1, 5, 10, 20 and 50 results, respectively. We conduct t-test to measure statistical significance of the results.

Table 2: Main results of experiments over short time gap datasets. Results marked with * are statistically significantly ($p < .1$) better than the best baseline GT. All the proposed methods significantly ($p < .01$) outperform the baseline HMM in both MRR and MAP.

(a) [2002,2007]→[1992,1996]

	MRR (Impr.%)	MAP (Impr.%)	P@1 (%)	@5 (%)	@10 (%)	@20 (%)	@50 (%)
NT	0.119 (-34.6)	0.136 (-39.9)	3.8	17.0	23.8	30.1	41.7
HMM	0.112 (-38.4)	0.145 (-36.0)	3.3	16.3	17.3	25.8	33.5
GT	0.182	0.226	7.6	27.2	33.7	50.0	63.0
CC	0.138 (-24.2)	0.188 (-17.2)	4.3	21.7	28.3	40.2	72.8
CC+SC	0.126 (-30.7)	0.174 (-23.2)	6.5	18.5	26.1	33.7	58.7
HT	0.213 (+16.9)	0.287 (+26.7)	8.7	32.6	42.4	58.7	77.2
HT+CC	0.231 (+27.1)	0.306 (+35.2)	**13.0**	33.7	**46.7**	56.5	71.7
HT+SC	0.235* (+29.0)	0.311* (+37.3)	13.0	**35.9**	45.7	54.3	**78.3**
HT+CC+SC	**0.238*** (+31.0)	**0.317*** (+40.1)	**13.0**	**35.9**	43.5	**60.9**	73.9

(b) [2002,2007]→[1987,1991]

	MRR (Impr.%)	MAP (Impr.%)	P@1 (%)	@5 (%)	@10 (%)	@20 (%)	@50 (%)
NT	0.129 (-27.8)	0.189 (-28.9)	4.5	17.9	21.4	28.4	35.8
HMM	0.101 (-43.5)	0.150 (-43.6)	4.0	11.1	16.4	22.2	31.1
GT	0.179	0.266	10.2	23.6	35.1	49.3	69.8
CC	0.141 (-21.0)	0.215 (-19.2)	5.3	20.9	30.7	53.8	74.7
CC+SC	0.147 (-17.9)	0.229 (-14.0)	6.2	20.9	30.2	47.1	76.0
HT	0.216* (+21.0)	0.349* (+31.4)	11.1	**32.4**	43.6	**63.1**	**78.7**
HT+CC	0.215* (+20.3)	0.345 (+29.8)	**12.4**	30.2	41.3	54.7	71.6
HT+SC	0.215* (+20.3)	0.345 (+29.9)	10.7	31.6	**46.7**	62.2	76.9
HT+CC+SC	**0.219*** (+22.7)	**0.353*** (+32.9)	**12.4**	30.2	40.9	55.6	75.1

▲ We show HT_{max}, $\{HT+CC\}_{max}$, $\{HT+SC\}_{max}$, $\{HT+CC+SC\}_{max}$, $\{CC+SC\}_{max}$, CC_{max} in this table as they all had better performance than the corresponding "sum" strategies.

7.2 Results Analysis

The evaluation results of our methods are summarized in Tab. 2 for the short and in Tab. 3 for long time separation. In addition, Tab. 6 presents several example results from the experiments on the short and long time gap datasets. The main observation is that most of our proposed methods statistically significantly outperform the most competitive baseline (**GT**) over both the short and long periods datasets. In the following subsections we first focus on the results over the short time gap datasets analyzing them from diverse perspectives, and then we discuss the results over the long time gap datasets as well as any performance differences resulting from the increase of the time gap.

7.2.1 Improving State-of-Art. As shown in Tab. 2a and 2b, the method **HT+CC+SC** statistically significantly ($p < .1$) outperforms the best baseline **GT** and also achieves significantly better results ($p < .01$) than another state-of-art baseline **HMM**. This is true for MRR and MAP measures for both the tested search periods of the short time gap datasets. The poorer performance of **HMM** when compared with methods that make use of word embedding spaces is because **HMM** is still a bag of words' based approach, albeit an improved one. Its assumption of little change in the terms' context cannot held any more, especially, when increasing the time gap between the base and target datasets (when contrasting its results in Tab. 2a and 2b).

Table 3: Main results of experiments over long time gap datasets. Results marked with * are statistically significantly ($p < .1$) better than the state-of-art baseline GT. † indicates those statistically significantly better than GT method ($p < .05$).

(a) [2004,2009]→[1967,1976]

	MRR (Impr.%)	MAP (Impr.%)	P@1 (%)	@5 (%)	@10 (%)	@20 (%)	@50 (%)
GT	0.061	0.103	3.3	7.7	12.0	14.3	26.4
CC	0.092 (+49.3)	0.165 (+60.5)	5.6	12.0	12.0	19.4	24.1
CC+SC	0.098 (+59.8)	0.175 (+70.5)	6.5	12.0	15.7	17.6	25.0
HT	0.117† (+89.8)	0.182* (+77.7)	6.5	**16.7**	**19.4**	**27.8**	**36.1**
HT+CC	0.102* (+66.7)	0.167 (+62.6)	6.5	11.1	13.9	23.1	30.6
HT+SC	**0.120**† (+94.6)	**0.204**† (+98.3)	**8.3**	13.0	17.6	25.0	34.3
HT+CC+SC	0.104* (+69.9)	0.179* (+74.4)	6.5	13.0	14.8	21.3	29.6

(b) [2004,2009]→[1939,1955]

	MRR (Impr.%)	MAP (Impr.%)	P@1 (%)	@5 (%)	@10 (%)	@20 (%)	@50 (%)
GT	0.054	0.083	0.0	13.9	18.5	21.3	28.7
CC	0.064 (+18.4)	0.070 (-15.5)	2.2	11.5	12.8	16.5	23.9
CC+SC	0.070 (+29.4)	0.102 (+23.3)	4.4	8.9	7.3	11.9	23.9
HT	**0.102**† (+89.1)	**0.156*** (+89.0)	**5.5**	**16.6**	**20.2**	**25.7**	30.3
HT+CC	0.078 (+44.9)	0.121 (+46.2)	3.3	14.0	17.4	21.1	29.4
HT+SC	0.099† (+82.7)	0.144* (+74.7)	**5.5**	**16.6**	17.4	22.9	**37.6**
HT+CC+SC	0.082* (+52.2)	0.123 (+49.1)	3.3	12.8	17.4	22.9	30.3

▲ We show HT_{sum}, $\{\text{HT+CC}\}_{\text{sum}}$, $\{\text{HT+SC}\}_{\text{sum}}$, $\{\text{HT+CC+SC}\}_{\text{sum}}$, $\{\text{CC+SC}\}_{\text{sum}}$, CC_{sum} in the table as they all achieved higher performance than the corresponding "max" strategy.
▲ For long time gap datasets, we only use the best performing baseline **GT**.

When comparing with baseline **GT**, we can observe on average 23% increase in MRR and 33% increase in MAP for all the four proposed methods. It confirms the previously mentioned hypothesis that the transformation biased on dual hierarchy helps to construct better mapping between terms in the two vector spaces.

7.2.2 Necessity of Transformation. The next observation is that method **NT** achieves relatively low performance (see Tab. 2). **NT** essentially assumes a static world in which every term is supposed to retain its semantics across the different domains (or should have the same "position" in a single joint vector space created on the merged set of documents from the different time periods). Yet, many terms change their meaning and usage in different times. Thus, their relative "positions" w.r.t. to other terms should change, too. Without the transformation, the information on the relative changes of term positions in the vector spaces is lost.

7.2.3 Necessity of Semantic Correspondence. Looking at the results in Tab. 2 and 3, we can observe that **CC** and **CC+SC** continue to under-perform all the proposed methods over both the short and long time gap datasets. Compared with our methods, on average, **CC+SC** is 27% lower in MRR, 29% lower in MAP, while **CC** has 30% lower results in MRR and 38% lower in MAP. This indicates that leveraging hierarchy/cluster information only is not enough and that the semantic transformation is necessary in our task.

7.2.4 Analysis of Different Signals for Across-Time Similarity Computation. In Sec. 5, we have proposed computing additional features: **CC** and **SC** for boosting the search performance. As seen in Tab. 2a, we can observe an increase of performance by incorporating the information about structural alignment together with the semantic transformation. **HT+CC** has 8.5% better performance than **HT** and **HT+CC+SC** achieves 3.3% better results than **HT+CC** when considering MRR. However when it comes to more distant past (see Tab. 2b), the impact of the structural alignment is decreasing. In fact, when testing over longer time gap datasets, incorporating **CC** and **SC** components even harms the performance: in Tab. 3a, **HT+SC** achieves better result than **HT+CC+SC** and in Tab. 3b, **HT** performs best. We can then conjecture that the "local" hierarchy structure changes as time passes.

7.2.5 Effect of Searching across Long Time Periods. When comparing the results in Tab. 3 and Tab. 2, we can observe a decrease in both MRR and MAP when performing search in the distant past. The reasons as we noticed is that many anchor terms changed semantics. The remedy is to decrease the number of anchor terms used in Eq. 1 as well as in Eq. 3[9]. Another observation is rather dramatic decrease in the performance of the baseline **GT** over the long time gap datasets (68% decrease in MRR and 90% decrease in MAP). Our method **HT** also returns worse results than for the short term gap datasets, yet, the performance drop is smaller (49% decrease in MRR and 47% decrease in MAP). In other words, the relative improvement of **HT** in relation to **GT** for the distant past is higher than that for the near past.

7.2.6 Effect of Different Queries. We analyze here the performance of the methods over different categories of queries distinguished by: (1) frequency in the datasets and (2) the query type[10].

Tab. 4 presents MRR scores for queries divided by their frequencies. To distinguish "frequent" from "infrequent" queries, we order queries by their frequencies in the base datasets in descending order. The ones in top 50% are regarded as "frequent" queries while the remaining half become "infrequent" ones. We see in Tab. 4 that the search task is "easier" for queries which are frequently mentioned in the datasets. All the methods achieve higher MRR for the frequent queries than for the infrequent ones over both the short and long time gap datasets. The proposed methods are capable of enhancing the performance of frequent queries by 68% and infrequent queries by 25%, on average, when compared to the results of **GT**.

Tab. 5 demonstrates the evaluation results categorized by query type: persons, locations and objects. Compared to the objects, searching for corresponding persons and locations across-time tends to be more difficult, resulting in lower MRR scores. It might be because objects are usually more specific and their contexts may be more fixed, hence, undergoing less variation. On the contrary, locations and persons usually appear in a large array of contexts (i.e., in many different circumstances, in relation to diverse types of events, diverse contexts, etc.). Comparing with **GT** our methods improve MRR score for queries in all the range of categories, especially, for locations, the average increase in MRR is 93%.

8 CONCLUSIONS

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 temporal document collections or can be helpful for

[9]To generate the results in Tab. 3b we used 3% of the top frequent anchor terms (as determined on the held-out calibration set) instead of 5% used for the other periods.
[10]Considering both the short and long time gap datasets.

Table 4: MRR scores for queries by frequency

	Short Time Gap		Long Time Gap	
	Frequent	Infrequent	Frequent	Infrequent
GT	0.203	0.156	0.053	0.028
HT	0.239	0.191	0.111	**0.050**
HT+CC	0.250	0.189	0.110	0.031
HT+SC	0.248	**0.193**	**0.121**	0.036
HT+CC+SC	**0.258**	0.191	0.112	0.026

Table 5: MRR scores for queries by query type

MRR	Persons (Impr.%)	Locations (Impr.%)	Objects (Impr.%)
GT	0.074	0.075	0.193
HT	**0.089** (+20.2)	0.138 (+85.0)	0.237 (+23.1)
HT+CC	0.084 (+14.2)	**0.149** (+99.1)	0.232 (+20.5)
HT+SC	0.087 (+18.0)	0.145 (+94.5)	0.240 (+24.8)
HT+CC+SC	0.083 (+13.2)	0.144 (+93.3)	**0.241** (+25.2)

automatically constructing evolution timelines and for clarifying word meaning. To find counterpart terms across time, we propose an advanced transformation technique based on dual hierarchical structures and on considering not only semantic but also the structural alignment. Through experiments we demonstrate that the proposed approaches outperform the state-of-art methods on both the short and long time gap datasets. We also illustrate in details the effectiveness of the three proposed signals as well as the limitations of the two structural signals over long time gap datasets. Thanks to the systematic and transparent character of our approach it is possible to understand the effect and role of each component on the quality of generated results. Finally, we note that our methods are fully unsupervised and work on a raw text without the requirement for any external ontologies, lexicons nor knowledge bases specific to any particular time periods. They can be thus applied to any combinations of time periods and underlying document collections.

In the future, we plan to investigate the way to detect temporal counterparts from particular viewpoints or particular senses. We also plan to extend the similarity computation to larger text units like sentences or documents.

ACKNOWLEDGMENTS

This research and development work was partially supported by JSPS Grant-in-Aid for Scientific Research (#15H01718, #17H01828, #15K12158) and the MIC/SCOPE (#171507010).

REFERENCES

[1] J. Aitchisonn. 2001. *Language Change, Progress or Decay?* Cambridge University Press.
[2] O. Alonso, M. Gertz, and R. Baeza-Yates. 2009. Clustering and Exploring Search rResults using Timeline constructions. In Proc. of CIKM, 97–106.
[3] O. Alonso and K. Shiells. 2013. Timelines As Summaries of Popular Scheduled Events *(In Proc. of WWW)*. 1037–1044.
[4] R. Baeza-Yates. 2005. Searching the Future *(In SIGIR Workshop MF/IR)*.
[5] K. Berberich, S. J. Bedathur, M. Sozio, and G. Weikum. 2009. Bridging the Terminology Gap in Web Archive Search *(In Proc. of WebDB)*.
[6] David M Blei and John D Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 113–120.
[7] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.
[8] John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain Adaptation with Structural Correspondence Learning. In *Proc. of EMNLP*. 120–128.
[9] L. Campbell. 2004. *Historical Linguistics, 2nd edition.* MIT Press.
[10] R. Campos, G. Dias, A. M. Jorge, and A. Jatowt. 2014. Survey of Temporal Information Retrieval and Related Applications. *Comput. Surveys* 47, 2 (2014), 15:1–15:41.
[11] W. Dakka, L. Gravano, and G. Ipeirotis. 2012. Answering general time-sensitive queries. *IEEE TKDE* 24, 2 (2012), 220–235.
[12] B. Falkenhainer, K. D. Forbus, and D. Gentner. 1989. The structure-mapping engine: Algorithm and examples. *Artificial intelligence* 41, 1 (1989), 1–63.
[13] D. Gentner. 1983. Structure-Mapping: A Theoretical Framework for Analogy*. *Cognitive science* 7, 2 (1983), 155–170.
[14] G. Hughes. 1988. *Words in Time: A Social History of the English Vocabulary*. Basil Blackwell.
[15] A. Jatowt and K. Duh. 2014. A framework for analyzing semantic change of words across time *(In Proc. of JCDL)*. 229–238.
[16] A. Jatowt and C. M. Au Yeung. 2011. Extracting collective expectations about the future from large text collections *(In Proc. of CIKM)*. 1259–1264.
[17] A. C. Kaluarachchi, A. S. Varde, S. Bedathur, G. Weikum, J. Peng, and A. Feldman. 2010. Incorporating Terminology Evolution for Query Translation in Text Retrieval with Association Rules *(In Proc. of CIKM)*. 1789–1792.
[18] N. Kanhabua and K. Nørvåg. 2010. Exploiting Time-based Synonyms in Searching Document Archives *(In Proc. of JCDL)*. 79–88.
[19] N. Kanhabua and K. Nørvåg. 2011. A comparison of time-aware ranking methods *(In Proc. of SIGIR)*. 1257–1258.
[20] Makoto P. Kato, Hiroaki Ohshima, Satoshi Oyama, and Katsumi Tanaka. 2010. Search As if You Were in Your Home Town: Geographic Search by Regional Context and Dynamic Feature-space Selection. In *Proc. of CIKM*. 1541–1544.
[21] Makoto P. Kato, Hiroaki Ohshima, and Katsumi Tanaka. 2012. Content-based Retrieval for Heterogeneous Domains: Domain Adaptation by Relative Aggregation Points. In *Proc. of SIGIR*. 811–820.
[22] Y. Kim, Y-I. Chiu, K. Hanaki, D. Hegde, and S. Petrov. 2014. Temporal Analysis of Language through Neural Language Models *(In Proc. of ACL Workshop)*. 61–65.
[23] V. Kulkarni, R. Al-Rfou, B. Perozzi, and S. Skiena. 2015. Statistically Significant Detection of Linguistic Change *(In Proc. of WWW)*. 625–635.
[24] T. Dumais L. Elsas. 2010. Leveraging Temporal Dynamics of Document Content in Relevance Ranking *(In Proc. of WSDM)*. 1–10.
[25] E. Lieberman, J. B. Michel, J. Jackson, T. Tang, and M. A. Nowak. 2007. Quantifying the evolutionary dynamics of language. *Nature* (2007), 713–716.
[26] D. Metzler, R. Jones, F. Peng, and R. Zhang. 2009. Improving Search Relevance for Implicitly Temporal Queries *(In Proc. of SIGIR)*. 700–701.
[27] R. Mihalcea and V. Nastase. 2012. Word Epoch Disambiguation: Finding How Words Change Over Time *(In Proc. of ACL)*. 259–263.
[28] T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proc. of ICLR Workshop*.
[29] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013. Distributed Representation of Phrases and Their Compositionality. In *Proc. of NIPS*. 3111–3119.
[30] S. Pan and Q. Yang. 2010. A survey on transfer learning. *IEEE TKDE* 22, 10 (2010), 1345–1359.
[31] M. Pargel, Q. D. Atkinson, and A. Meade. 2007. Frequency of word-use predicts rates of lexical evolution throughout Indo-European history. *Nature* 449 (2007), 717–720.
[32] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. *Learning internal representations by error propagation.* Technical Report. California Univ, San Diego La Jolla Inst. For Cog-nitive Science.
[33] E. Sandhaus. 2008. The New York Times Annotated Corpus Overview. *The New York Times Company, Research & Develop.* (2008), 1–22.
[34] N. Tahmasebi, G. Gossen, N. Kanhabua, H. Holzmann, and T. Risse. 2012. NEER: An Unsupervised Method for Named Entity Evolution Recognition *(In Proc. of Coling)*. 2553–2568.
[35] N. K. Tran, A. Ceroni, N. Kanhabua, and C. Niederée. 2015. Back to the Past: Supporting Interpretations of Forgotten Stories by Time-aware Re-Contextualization *(In Proc. of WSDM)*. 339–348.
[36] Peter D. Turney. 2006. Expressing Implicit Semantic Relations without Supervision. *CoRR* (2006).
[37] Peter D Turney. 2008. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research* (2008), 615–655.
[38] Xuerui Wang and Andrew McCallum. 2006. Topics over time: a non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 424–433.
[39] Y. Zhang, A. Jatowt, S. S. Bhowmick, and K. Tanaka. 2015. Omnia Mutantur, Nihil Interit: Connecting Past with Present by Finding Corresponding Terms across Time. In *Proc. of ACL*. 645–655.
[40] Yating Zhang, Adam Jatowt, Sourav S Bhowmick, and Katsumi Tanaka. 2016. The Past is Not a Foreign Country: Detecting Semantically Similar Terms across Time. *IEEE Transactions on Knowledge and Data Engineering* 28, 10 (2016), 2793–2807.
[41] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. 2005. *Time-sensitive Dirichlet process mixture models.* Technical Report. DTIC Document.

Table 6: Example results for the short and long time gap datasets. For each query ("P" for persons, "L" for locations and "O" for objects), we show the returned lists by the best performing proposed methods for each time period and by the best performing baseline GT. Bold font indicates the ground truth, and the numbers in parenthesis are the ranks of the ground truth answers within the returned list.

[2002,2007]→[1992,1996]		
Query	**Ranking List Returned by HT+CC+SC**	**Ranking List returned by GT**
Merkel (P)	Kwasniewski, Balladurs, Kono, Chirac . . . **Chancellor Kohl (13)**	Balladurs, Meri, Kebich, Zepeda . . . **Chancellor Kohl (149)**
Annan (P)	**Boutros Boutrosghali (1)**, Juppe, Krajisnik, Holbrooke, Silajdzic	**Boutros Boutrosghali (1)**, Juppe, Krajisnik, Holbrooke, Silajdzic
Aznar (P)	Silvio Berlusconi, Lamberto Dini, Hun Sen, Jeanluc Dehaene . . . **Felipe Gonzalez (8)**	Hun Sen, Prince Ranariddh, Kuchma, Javier Solana . . . **Felipe Gonzalez (179)**
Slovakia (L)	Romania, Portugal, Bulgaria . . . **Slovak Republic (109)** . . . **Czechoslovakia (115)**	Romania, Portugal, Bulgaria . . . **Slovak Republic (131)** . . . **Czechoslovakia (186)**
Czech (L)	Hungary, Bulgaria, Austria . . . **Czechoslovakia (11)** . . . **Czech Republic (158)**	Hungary, Bulgary, Austria . . . **Czechoslovakia (11)** . . . **Czech Republic (167)**
Berlin (L)	Paris, Vienna, Budapest . . . **Berlin (30)** . . . **Bonn (64)**	Paris, Vienna, Budapest . . . **Berlin (35)** . . . **Bonn (62)**
iPod (O)	**laptop (1)**, adapter, desktop, **CD player (4)** . . . **PDA (20)**	adapter, deskop, **CD player (3)** . . . **laptop (5)** . . . **PDA (20)**
Firefox (O)	web_browser, browser, **Netscape Navigator (3)**, **Netcruiser (4)**, **Internet Explorer (38)**	web browser, bjc600 . . . **Netcruiser (14)**, **Netscape Navigator (21)**, **Internet Explorer (35)**
Nato (O)	**Nato (1)**, natoled peacekeeping, nationalist serb, krajina serb	nationalist serb, hezbollah guerrillas, krajina serb . . . **Nato (7)**
[2002,2007]→[1987,1991]		
Singh (P)	Singh, Jatoi, **Gandhi (3)**, Junejo, Chandra shekhar	Pollino, Lopez, Scopino, Rodriguez . . . **Gandhi (226)**
Jacques Rogge (P)	IOC, Mike Jacki, Robert Helmick, Werner Fricker . . . **Samaranch (14)**	Mike Jacki, Robert Helmick, Marat Gramoc, Werner Fricker . . . **Samaranch (92)**
Vladimir Putin (P)	**Boris Yeltsin (1)**, Leonid Brezhnev, Helmut_Kohl, Francois Mitterrand . . . **Mikhail Gorbachev (9)**	**Boris Yeltsin (1)**, Leonid Brezhnev, Helmut Kohl, Francois Mitterrand . . . **Mikhail Gorbachev (9)**
Myanmar (L)	Thailand, Pakistan, Bangladesh . . . **Burma (50)**, **Myanmar(51)**	Thailand, Pakistan, Bangladesh . . . **Burma (50)** . . . **Myanmar (86)**
Czech (L)	**Czechoslovakia (1)**, Slovakia, Netherlands, Sweden Finland, Scandinavian countries	Netherlands, Sweden Finland, **Czechoslovakia (3)** . . . Belgium, Finland,
Macedonia (L)	Rwanda, Albania, Myanmar, Tirana . . . **Yugoslavia (53)**	Rwanda, Albania, Myanmar, Tirana . . . **Yugoslavia (95)**
Boeing (O)	**Airbus (1)**, **Mcdonnell Douglas (2)**, airbus industry, **Boeing (4)**, Taiwan Aerospace	**Boeing (1)**, 777s, Daimler, **Airbus (6)** . . . **Mcdonnell Douglas (12)**
Linux (O)	**Unix (1)**, software, **OS2 (3)**, industrystandard . . . **MSDOS (30)**	software, industrystandard, **Unix (3)** . . . **MSDOS (52)** . . . **OS2 (62)**,
spreadsheet (O)	**Lotus 123 (1)**, **database (2)**, wordprocessing . . . **spreadsheet (19)** . . . **quattro pro (43)**	**database (1)**, wordprocessing . . . **spreadsheet (13)** . . . **quattro pro (72)** . . . **Lotus 123 (110)**
[2004,2009]→[1967,1976]		
Query	**Ranking List Returned by HT+SC**	**Ranking List returned by GT**
Schröder (P)	Wagg, Scliroder, Grossart, Whitburgh . . . **Brandt (14)**	Munchmeyer, Rmani, Turban, Whitburgh . . . **Brandt (485)**
Chirac (P)	**Pompidou (1)**, Mitterrand . . . **Poher (25)**, **d'Estaing(26)** . . . **Gaulle (793)**	Tsatsos, **Pompidou (30)** . . . **Poher (604)**, **d'Estaing(479)** . . . **Gaulle (929)**
Medvedev (P)	Filbinger, **Brezhnev (2)**, Podgorny, Lunkov, Alinister	**Brezhnev (1)**, Sayem, Chissano, Filbinger, Karamnanlist
Zapatero (P)	Rodriguez, Portillo, Seinor, Jumenez . . . **Suarez (19)**	Garrigues, Concalves, Calazans, Portillo . . . **Suarez (46)**
Mumbai (L)	**Bombay (1)**, Karachi, Delhi, Dacca, Delbi	Jakarta, Kualalumpur, Manila, Rangon . . . **Bombay (11)**
Sri Lanka (L)	**Sri Lanka (1)**, **Ceylon (2)**, India, Colomo, Pakistan, Indonesia	Fiji, Pakistan, Madagascar . . . **Sri Lanka (22)** . . . **Ceylon (126)**
Myanmar (L)	**Burma (1)**, Thailand, Laos, Indonesia, Guam	Yemen, Indonedia, Thailand, Maldives . . . **Burma (102)**
Berlin (L)	Munich, Berlin, Moscow, Vienna . . . **Bonn (21)**	Prague, Berlin, Moscow, Munich . . . **Bonn (485)**
[2004,2009]→[1939,1955]		
Query	**Ranking List Returned by HT**	**Ranking List returned by GT**
Putin (P)	Kremlin, Kabanov . . . **Malenkov (4)** . . . **Khrushchev (16)** . . . **Stalin (55)**	Zorin, Vinogradov, **Malenkov** (22) . . . **Stalin (321)** . . . **Khrushchev (353)**
Kwasniewski (P)	Celal, President, Chamoun . . . **Raczkiewicz (62)** . . . **Bierut (709)**	Omukama, Mostras, Celal . . . **Raczkiewicz (184)** . . . **Bierut (747)**
Berlusconi (P)	minister . . . **Segni (8)** . . . **Scelba (120)** . . . **Pella (266)** . . . **Gasperi (441)**	signor **Segni (24)** . . . **Scelba (247)** . . . **Pella (507)** . . . **Gasperi (632)**
Koizumi (P)	minister . . . **Katayama (3)** . . . **Hatoyama (12)** . . . **Yoshida (350)** . . . **Shidehara (544)**	minister . . . **Katayama (4)** . . . **Hatoyama (28)** . . . **Yoshida (516)** . . . **Shidehara (778)**
Mumbai (L)	**Bombay (1)**, Delhi, Karachi, Rangoon, Beirut	Delhi, Panjim, Rangoon, Cairo . . . **Bombay (9)**
Russia (L)	Poland, Finland, Czechoslovakia, Bulgaria . . . **Soviet (61)**	Finland, Yugoslavia, Poland, Czechoslovakia . . . **Soviet (159)**
Sri Lanka (L)	India, Colombo, Ahmsd, Indonesia . . . **Ceylon (6)**	India, Colombo, pakistan, Indonesia . . . **Ceylon (289)**
Thailand (L)	Indonesia, Philippines, **Siam (3)**, Malaysia, Manila	China, Hongkong, Laos, **Siam (12)** , Malaysia