Finding "Similar" Concepts with Evidences across Different Feature Vector Spaces

Yating ZHANG  Adam JATOWT  and  Katsumi TANAKA
Graduate School of Informatics, Kyoto University Yoshida Honmachi, Sakyo, Kyoto 606-8501, Japan
E-mail: {zhang, adam, tanaka}@ dl.kuis.kyoto-u.ac.jp

Abstract  “The past is a foreign country: they do things differently there” is an often quoted opening sentence of a famous novel by L. Hartley entitled “Go Between”. It emphasizes the well-known observation that it is often difficult to understand entities, concepts and their context in the past, especially, in the distant past. Even though some professionals, common sense or just our intuition might suggest that certain entities may be similar across time, still, usually, we lack convincing and concrete evidence to support similarity estimation. For example, it may not be immediately obvious why Walkman in 1980s is considered to be a similar entity to iPod. However, we can easily understand their similarity when learning that both were dominant, portable music devices in their corresponding times. In this paper, we propose to automatically detect evidences to explain similarities and differences between entities in different time periods. For a given input entity we first output the ranked list of candidate counterparts and then we detect supporting evidence to explain why the results are similar or different to the input entity. The evidences our method generates should be relevant to entities, cover their diverse and important aspects and allow for easy comparison.

Keyword  Evidence Finding, Similarity Searching, Difference Searching, Entity Comparison

1. INTRODUCTION

In the current fast-paced world, people tend to possess limited knowledge about things from the past. An average person typically knows only about events and entities taught at school or ones curated as collective memory - the highly selective representation of the past maintained by mass media. Furthermore, besides many past entities and events, also the contexts of distant times remain unknown for many users who do not actively study the past. Yet, maintaining the knowledge and the memory of the past is definitely important and, therefore, various memory institutions such as libraries and archives offer open access to past documents they keep. However, searching within such collections as well as understanding the retrieved content is hampered due to our limited knowledge of vocabulary used in the past and its meaning. Considering the limited knowledge of the past average users possess, it would be beneficial if the users were provided with some kind of assistance when interacting with archival document collections. Ideally, such assistance should empower them to search and comprehend within the past collections as efficiently as they would in “present” collections such as the Web.

In our previous work, we approached a research problem of temporal counterpart search which returns semantically similar terms from the past to an input query from the present time using large temporal document collections. [49]. In such an approach, a series of questions can be answered, such as “what music device 30 years ago played similar role as iPod does nowadays?” or “who today’s Beatles are?” However, answering “what” questions is not enough for users to deeply understand and verify the provided answers, such as to determine in which way a given temporal counterpart is similar to the queried entity or how different both of them are.

Considering the issues raised above, we focus in this work on solving the “why similar” question which helps explaining temporal counterparts by outputting “evidence” terms for clarifying the similarity and differences between the counterparts. For example, given a pair of entities (e.g., iPod and Walkman) considered as temporal counterparts, the system should return a set of evidences indicating (1) similar concepts maintained over time (i.e., both entities are used to listen to music, both are designed to be a portable music device, and both utilize some storage media to store songs); (2) significant differences that can be used to differentiate each other (i.e., iPod has large storage size enabling to store more songs, iPod allows watching movies, iPod has a display panel to show the information about songs such as lyrics, singer name and song name).

The problem of similarity and difference detection for temporal counterparts is not trivial. The key difficulty comes from the need for comparisons on the concept level
instead on the literal word level, since the context tends to change much across time, especially, over longer distances. In other words, for any pair of temporal counterparts, their context can be quite different due to time passage. Many cases that are in fact desired similar concepts between temporal counterparts may still in fact remain but may not be mentioned in either of the context. For example, iPod and Walkman both utilize storage media to store the songs, but in their context, the concept, storage media, might not be explicitly mentioned. Instead, for iPod, MP3 is considered as a suitable storage media, and Walkman uses cassette to store music. Note that naturally it is possible that the concepts (e.g., listening to music, portable music device) shared by the temporal counterparts are directly mentioned and thus can be easily detected by taking the overlap of their context. Our system however focuses on solving the former hard case where the concepts are not directly mentioned in the context.

Another challenge lies in the criteria necessary for selecting an informative set of evidences. First, it is important to select important aspects of entities. There may be many similar and different points between the compared entities, however, it is essential to extract only significant aspects and eliminate the obvious ones (i.e., both of iPod and Walkman are of rectangular shape). Secondly, the system should output organized results to offer useful explanation. Since the output is in the form of a set of evidences, it should be constructed considering the coverage and diversity of the selected items.

In view of the challenges mentioned above, we first propose an approach, which enables to compare terms in regards to their semantic meaning. In other words, we propose the comparison on a concept level, by bridging term representations from one vector space (e.g., one derived from the present documents) and those from another vector space (e.g., one built from the past documents). Terms in both the vector spaces are represented by the distributed vector representation [35,36]. Next, we propose methods to detect similar concepts maintained by the given two entities and to discover essential differences between them. Finally, we introduce an optimization function to optimize the returned set of similar and dissimilar points by the criteria of their coverage and diversity.

The rest of the paper is organized as follows. In the next section we review the related work. In Section 3 we formally define the problem and describe its background. We next describe the method for term representation and term comparison across time in Section 4. Then, we introduce our approaches for similarity and difference detection in Section 5. Section 6 describes the optimization way to organize the output. The experimental results are illustrated in Section 7. We conclude the paper and list future directions in Section 8.

2. RELATED WORK

Temporal Information Retrieval has become the subject of multiple studies in the recent years. Prior research focused on tasks such as time-aware document ranking [6,13,15,25,30], temporal organization of search results [2,3], query understanding [13,33,23], future in-information retrieval [4,19,21], analysis of how the word meanings change over time [20,27,28,34], or addition of context to explain the past [46] and so on.

Among the above topics, time-driven change of the semantic meaning - an emerging topic of study within historical linguistics [1,11,18,29] is relevant to this work. Several researchers employed computational methods for analyzing changes in word senses over time. Mihalcea et al. [34] classified words to one of three past epochs based on word contexts. Kim et al. [27] and Kulkarni et al. [28] computed the degree of meaning change by applying neural networks for word representation. Our objective is different from the above approaches as we directly search for corresponding terms across time, and, in our case, temporal counterparts can have different syntactic forms.

Certain works approached the problem of computing term similarity across time [5,22,24,45]. Kalurachchi et al. [22] proposed to discover semantically identical temporally altering concepts by applying association rule mining, assuming that concepts referred by similar events (verbs) are semantically related. Khabaua et al. [24] investigated detection of the change of terms through the comparison of temporal Wikipedia snapshots. Berberich et al. [5] approached the problem by introducing a HMM model and measuring the across-time semantic similarity between two terms by comparing the contexts captured using co-occurrence measures. Tahmasebi et al. [45] improved that 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. Several important differences distinguish our work from those works. First, the previous works focused mainly on detecting changes in 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, the above mentioned 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 (i.e., having same surface forms of context terms) but to have the same meaning.

Transfer Learning [39] is to some extent related to our work. It has been mainly used in tasks such as POS tagging [9], text classification [7,32,47], learning to rank [10,16,48] and content-based retrieval [26]. The temporal correspondence problem can also be understood as a transfer learning as it is a search process that uses samples in the base time for inferring correspondent instances 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.

3. BACKGROUND AND PROBLEM DEFINITION

In this section, we formally define the problem of similarity/difference search between two potentially similar entities.

**PROBLEM STATEMENT.** Given two potentially similar entity names, \( e_1 \) and \( e_2 \) where \( e_1 \) and \( e_2 \) exist in different spaces \( T_1 \) and \( T_2 \), respectively, the task is to find (1) the set of similar concepts they share, \( \text{Sim}(e_1,e_2) = \{w_i=w_{i0}, \ldots, w_i=w_{i\theta}\} \) where \( w_i \) and \( w_{i\theta} \) exist in \( e_1 \) and \( e_2 \) respectively; (2) the set of their differences, \( \text{Diff}(e_1,e_2) = \{w_i(e_1), \ldots, \theta_{i\theta}(e_2)\} \) where \( w_i(e_1) \) exists in \( e_1 \) but not in \( e_2, \theta_{i\theta}(e_2) \) denotes the opposite.

**DEFINITION 1 (SEMANTICAL SIMILARITY).** If the context of \( w \) is semantically similar to the context of \( \omega \), then \( w \) is semantically similar to \( \omega \).

4. ACROSS-TIME TERM COMPARISON

4.1 Term Representation

Distributed representation of words using neural networks was originally proposed by [42]. Mikolov et al. [35,36] 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 several advantages: (1) it captures precise semantic word relationships; (2) it can easily scale to millions of words. After applying Skip-gram model, a \( mxp \) matrix is created from the documents in one space (e.g., the documents in 2000s), \( D(T_1) \), where \( m \) is the vocabulary size and \( p \) are the dimensions of feature vectors. Similarly, a \( nxq \) matrix is constructed from the documents in another space (e.g., the documents exist in 1980s), \( D(T_2) \) (as shown in Fig. 2).

Fig. 2. Creating word vector representations for two spaces.

4.2 Term Comparison across Vector Spaces

Our goal is to compare words in two vector spaces. However, it is impossible to directly compare words in two different semantic vector spaces as the features (dimensions) in both spaces have no direct correspondence between each other (as shown in Fig. 1). To solve this problem, we train a transformation matrix to build the connection between the two vector spaces. To better imagine the transformation idea, the semantic spaces could be compared to buildings. If we regard two semantic spaces as two buildings, then, in order to map the components from one building to ones in the other one, we need first to know how the main frames of the two buildings correspond to each other. Afterwards, the rest of the components can be mapped automatically by considering their relative positions to the main frames of their building. So, in our case, having found the correspondence between the anchor terms in the two semantic spaces, we can automatically map all other remaining terms relative to these anchors. Fig. 3 conceptually portrays this idea by showing that the correspondence of anchor terms enables mapping other terms, such as iPod to Walkman so that the relative position between iPod and anchors in one space is similar to the relative position between Walkman and the corresponding anchors in another space (only two dimensions are shown.

![Fig. 1. Conceptual view of pair detection.](image1.png)

![Fig. 2. Creating word vector representations for two spaces.](image2.png)

![Fig. 3. Conceptual view of the transformation.](image3.png)
for simplicity).

![Conceptual view of the across vector spaces transformation by matching similar relative geometric positions in each space.](image)

For realizing the above described transformation, it is essential to first find good anchor terms (“main frames”) which can help to build the correspondence between any two semantic spaces. However, it is non-trivial to manually prepare large enough sets of anchor terms that would cover various domains as well as exist in any possible combinations of the base and target time periods. We then rely here on an approximation procedure for automatically providing anchor pairs. We select terms which (a) have the same syntactic forms in the base and the target time periods, and (b) which are frequent in both the time periods. Such Common Frequent Terms (CFTs) are then used as the anchor terms. One reason to choose CFTs as anchors is that frequent terms tend to be “connected” with many other terms. Another is that frequent terms (e.g., sky, river, music, cat) change their meanings over time only to small extent. The more frequently a word is used, the harder is to change its dominant meaning (or the longer time it takes for a word to undergo the meaning shift) as the word is commonly used by many people. The phenomenon that words used commonly in everyday language had evolved more slowly than words used less frequently has been observed in several languages including English [31,40]. This assumption guarantees relatively good correspondence between the two frames that would be “constructed” with the help of CFTs.

After determining the anchor terms, our task is to build the correspondence between two semantic spaces by utilizing the set of prepared anchor terms. In particular, we will train the transformation matrix to automatically map dimensions of the base vector space to the ones in the target vector space. Let us suppose there are \( K \) pairs of anchor terms \( \{(o_1, w_1), \ldots, (o_K, w_K)\} \) where \( o_i \) is an anchor in one space and \( w_i \) is its counterpart anchor in another space. The transformation matrix \( M \) is then found by minimizing the differences between \( M \cdot o \) and \( w_i \) (see Eq. 1). This is done by ensuring that the sum of Euclidean 2-norms between the transformed query vectors and their counterparts is as small as possible when using \( K \) anchor pairs. Eq. 1 is used for solving the regularized least squares problem (\( \gamma \) equals to 0.02) with the regularization component added to prevent overfitting.

\[
M = \arg \min_M \sum_{i=1}^{K} [ M \cdot o - w_i ]^2 + \gamma |M|^2
\]

5. SIMILARITY AND DIFFERENCE DETECTION

In this section, we first discuss the criteria of a set of terms to be useful for indicating the actual similarities or differences of terms across time. Based on these criteria, we then propose several features to extract similarities between two given entities and their differences.

5.1 Criteria for Selecting Evidences

**Similarity of entities.** \( <w_i, o_h> \) is a pair of terms where \( w_i \) appears in the context of entity \( e_1 \) and \( o_h \) occurs in the context of entity \( e_2 \). \( <w_i, o_h> \) explains how \( e_1 \) is similar to \( e_2 \). \( <w_i, o_h> \) should have at least some of the following characteristics: (a) \( w_i \) is highly relevant to \( e_1 \) while \( o_h \) is relevant to \( e_2 \); (b) \( w_i \) and \( o_h \) should indicate similar concept; (c) the relation between \( w_i \) and \( e_1 \) should be similar to the relation between \( o_h \) and \( e_2 \).

**Differences between entities.** \( w_i \) is a context term of entity \( e_1 \), denoted as \( w(e_1) \) such that it denotes a concept existing in \( e_1 \) but not in \( e_2 \). \( w(e_1) \) can explain the difference of \( e_1 \) from \( e_2 \). Similarly, \( o_h \) is a context term of entity \( e_2 \), denoted as \( o_h(e_2) \) such that it represents a concept existing in \( e_2 \) but not in \( e_1 \). \( o_h(e_2) \) can then explain the difference of \( e_2 \) from \( e_1 \). \( w_i \) (or \( o_h \)) should at least satisfy some of the following characteristics: (a) \( w_i \) (or \( o_h \)) is highly relevant to \( e_1 \) (or \( e_2 \)); (b) the concept behind \( w_i \) (or \( o_h \)) is less likely to appear in \( e_1 \) (or \( e_2 \)); (c) it is rare to find such relation of \( w_i \) and \( e_1 \) in the context of \( e_2 \).

5.2 Feature Estimation

We start to quantify the objectives listed above. Three features can be generalized from the above high-level criteria: **relevance**, **intra-similarity**, and **relational-similarity**. We define and estimate each feature as follows.

**Relevance.** \( Rel(w,e) \) is the strength of the relatedness between a context term \( w \) and entity \( e \). It is measured by the multiplication of two conditional probabilities (see Eq. 2). The left side guarantees the context term \( w \) is relevant to the entity \( e \) while the right side gives more priority to those terms which only co-occur with \( e \). Intuitively, the context term which is unique to the entity will have a high relevance score. For example, although the context term **music** frequently appears within the context of **iPod**, it
is not unique to iPod since music can co-occur with many other entities. On the other hand, Apple is more relevant to iPod since both frequently co-occurred with each other.

\[
rel(w, e) = P(w|e) - P(e|w)
\]

\[
= \frac{P(w,e)}{P(e)} - \frac{P(w, e)^2}{P(w)} = \left( \frac{n_{we}}{\max(n_{wi})} \right) \frac{\max(n_{ei})}{n_e} \tag{2}
\]

Note that Eq. 2 computes the relevance between each context term of a given input entity, e.g., \(rel(w_i, e_1)\) and \(rel(w_i, e_2)\) estimate the relevance of context terms of entity \(e_1\) and \(e_2\), respectively. To apply this feature for detecting the evidences to indicate the similarity between two entities, the relevance of each evidence, \(rel(<w_i, o_i>)\), can be computed as Eq. 3.

\[
rel(<w_i, o_i>) = rel(w_i, e_1) - rel(o_i, e_2) \tag{3}
\]

**Intra-Similarity.** This feature measures the similarity between a pair of context terms \(<w_i, o_i>\) where \(w_i\) is the context term of entity \(e_1\) and \(o_i\) is the context term of entity \(e_2\). As discussed in Sec. 4, since \(w_i\) and \(o_i\) come from different time periods (i.e., different vector spaces) to compare their similarity, it is essential to first transform the representation of a term (e.g., \(w_i\)) from one vector space (e.g., \(T_i\)) to another space (e.g., \(T_2\)). We then compare the transformed representation (e.g., \(M^i \cdot w_i\)) with the representation of the term in \(T_2\) (e.g., \(o_i\)) by cosine similarity. The intra-similarity of \(<w_i, o_i>\) is computed by Eq. 4.

\[
simIntra(<w_i, o_i>) = \cos(M \cdot w_i, o_i) \tag{4}
\]

**Relational-Similarity.** Besides the intra-similarity discussed above which measures the semantic similarity between \(w_i\) and \(o_i\), the relational similarity estimates if the relation between \(w_i\) and \(e_1\) is corresponds to the relation between \(o_i\) and \(e_2\). To represent the relation between \(w_i\) and \(e_1\), we take the difference of their vector representations, \(w_i - e_1\). The relational similarity is defined as below.

\[
simR(<w_i, e_1>, <o_i, e_2>) = \cos(M \cdot (w_i - e_1), (o_i - e_2)) \tag{5}
\]

---

**6. EVIDENCE FINDING**

Based on the features computed in Sec. 5, in this section, we introduce our method to detect a set of evidences indicating similarity (\(S_{sim}\)) and a set of evidences denoting the difference (\(S_{diff}\)) between two entities.

We define a good set \(S_{sim}\) (or \(S_{diff}\)) as the one in which each item have high quality but is different from each other. In other words, a good set of evidence should consider both quality and diversity within the set.

**6.1 Quality of Evidence**

We define the quality of the item in \(S_{sim}\) as the pair of terms \(<w_i, o_i>\) which should have high in relevance score, high in intra similarity and high in relational similarity (see Eqs. 6-7).

\[
qua_{sim}(<w_i, o_i>) = rel(<w_i, o_i>)^\alpha \cdot sim(<w_i, o_i>)^{1-\alpha} \tag{6}
\]

\[
sim(<w_i, o_i>) = simIntra(<w_i, o_i>) \cdot simR(<w_i, e_1>, <o_i, e_2>) \tag{7}
\]

On the other hand, the quality of the item in \(S_{diff}\) is given to a single term, \(w_i(e_1)\), which must be high in relevance score to \(e_1\), low in the intra similarity with all the context terms \((o_i)\) in \(e_2\) and low in the relation similarity (see Eq. 8).

\[
qua_{diff}(w_i(e_1)) = rel(w_i, e_1)^\alpha \cdot (1 - \max_{o_i \in e_2} \sim(<w_i, o_i>)^{1-\alpha} \tag{8}
\]

where, \(C\) is the set of context terms of \(e_2\). \(sim(<w_i, o_i>)\) is computed by Eq. 7.

**6.3 Output Optimization**

As mentioned before, diversity is an important criteria for a good set of evidences. In this section we propose an objective function to optimize the output evidences by considering both the quality and diversity of the returned set.

![Diagram of the optimization process](image)

1. The evidence should be relevant to its entity (relevance)
2. \(w_i\) and \(o_i\) should be similar to each other (similarity)
3. The selected pairs of evidences should be different from each other (diversity)

Eq. 9 is used for the process of finding good set. Each time we select a high quality item which, at the same time, should be as dissimilar as possible to the already selected items, until reaching a predefined size of returned (selected) set.

\[
L = \max_{A, A_2} \{ \lambda \cdot qua(A) - (1 - \lambda) \max_{A, A_2} \{simInter(A, A_2)\} \} \tag{9}
\]

\[
simInter(A, A_2) = \cos(<w_i, o_i>, <w_i, o_i>) \tag{10}
\]

\[
simInter(A, A_2) = \cos(w_i, w_i) \tag{11}
\]

for \(S_{sim}\),

\[
simInter(A, A_2) = \cos(w_i, w_i) \tag{12}
\]

for \(S_{diff}\).
$L$ is the size of the returned set. $S$ denotes the selected set and $A_i$ is an item in $S$, while $U$-$S$ indicates the unselected set and $A_i$ is the item in $U$-$S$. Note that Eq. 9 is a generic function which can be applied to optimize the evidences $S_{\text{union}}$ when replacing $g(u)(A_i)$ by Eq. 6 or to optimize the evidences $S_{\text{diff}}$ by substitution with Eq. 8.

7. EXPERIMENTS
7.1 Dataset
For the experiments we use the New York Times Annotated Corpus [43]. This dataset contains over 1.8 million newspaper articles published between 1987 and 2007. We select two time periods [1987,1991] and [2002,2007] for testing our proposed methods on finding similarity/difference evidences between two entities across time (e.g., Walkman from [1987,1991] and iPod appears in [2002,2007]). Each time period contains around half a million news articles. We next train the model of distributed vector representation separately for these two time periods. The vocabulary size of the entire corpus is 360k, while the vocabulary size of each time period is around 300k.

7.2 Test Sets
To the best of our knowledge, there are no standard test benches for our research task. We plan then to resort to manual construction of test sets. The test sets will contain two entity names with two sets of evidences respectively indicating similarity and difference between them. In order to test the performance of the proposed methods over different types of queries, we are going to categorize the tested pairs of entities into three groups: $(A,A)$, $(A,A')$ and $(A,B)$. $(A,A)$ represents those queries where two components of query have the same surface form such as (Japan, Japan). Here the user’s search intent might be detecting the consistency and changes of the same entity across time (e.g., comparison between Japan in [2002,2007] and in [1987,1991]). $(A,A')$ denotes queries where two entities have different surface forms but are similar in semantics, such as (iPod, Walkman). In our previous work [49], we named them as temporal counterparts. For the purpose of experiments, we will select several temporal counterparts that we used before¹ as tested queries. Finally, $(A,B)$ is the type of queries where $A$ and $B$ are totally different entities, that is, both are different in surface form and in semantic meaning (or there is no correspondence between them). Users may search using such queries when they are curious about any evidences that can indicate similarity between two entities, which are typically considered different. To create the test sets we will utilize external resources including the Wikipedia, a Web search engine and several historical textbooks. In total, 51 pairs of entities will be tested. All the test pairs are going to be made available for others to experiment with.

We are going to add the experimental results over the manual constructed test sets in the camera ready paper. As for now, we show some of the examples of the test queries ($e_1$, $e_2$) and their results in Table 1 ($e_1$ denotes an entity existing in [2002,2007] and $e_2$ is the entity in [1987,1991]).

7.3 Evaluation Measures and Tested Methods
Evaluation Measures. We will use precision and recall as the main measures for evaluating the returned set of evidences.

Baselines. We are also going to prepare two baselines as follows:

1. Overlap approach (Overlap): this method detects similarities by directly computing the overlap of the context of two queried entities and selecting the most relevant ones; the differences (e.g., $w_i(e_1)$) will be extracted from non-overlapped terms with the condition that the terms (e.g., $w_i$) should be highly relevant to their corresponding entity (e.g., $e_1$). We will test Overlap approach to examine whether the distributed vector representation and transformation are necessary.

2. Skip-gram Model without transformation (COM): the purpose of including this baseline is to test the necessity of transformation across vector spaces for detecting good sets of words denoting similarities and differences. Since COM also uses distributional representation for capturing word semantics same as the proposed methods do, the only difference is that the term vector training process combines the documents from two time periods. Thus the vocabularies in the two time periods are represented within one vector space. Different from the separate training process, the combined training will likely lose the relative positions between the terms in each time period. Also COM assumes that the terms existing in both time periods have exactly same meaning (or does not substantially change their meanings) since there is a unique vector representation for each term in the combined vector space.

Proposed Methods. We will test the proposed methods

¹ http://www.dl.kuis.kyoto-u.ac.jp/~adam/temporalsearch_short.txt
mentioned in the previous sections. All of them use the neural network based term representation. Since our methods consider several features (relevance, intra-similarity and relational-similarity) in the experiments we will not only compare our proposed methods with all the features included, but we are also going to test the effectiveness of each feature by adjusting the weight of each feature and tracking performance variations.

7.4 Experimental Results

Table 1 displays the returned similarity evidences for few example queries. The returned size of evidence equals 20 in the experiments.

8. Conclusion and Future Work

In this paper we have proposed a method for computing similarities and differences between pair of entities across time. The objective is to help users understand how things or concepts in the past differ from the ones existing at present. We have applied transformation matrix and considered several characteristics of ideal evidences of similarities and differences such as relevance, similarity and diversity.

In the future we plan to conduct extensive experiments to evaluate our method over diverse datasets and different lengths of time periods. In addition, we will design more exhaustive approaches such as ones using clustering.

Table 1. Example results of similarity evidences generated by the proposed methods where ei is the entity in the time [2002,2007] and ej is in [1987,1991]. In the experiments, the size of the returned evidence set equals 20.

<table>
<thead>
<tr>
<th>(ei, ej)</th>
<th>(ipod, walkman)</th>
<th>(apple, sony)</th>
<th>(putin, yeltsin)</th>
<th>(facebook, use net)</th>
<th>(japan, japan)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;apple, sony&gt;</td>
<td>&lt;ipod, sony&gt;</td>
<td>&lt;russian, michael&gt;</td>
<td>&lt;myspace, unix&gt;</td>
<td>&lt;koizumi, kaifu&gt;</td>
</tr>
<tr>
<td>2</td>
<td>&lt;itunes, cassettes&gt;</td>
<td>&lt;itunes, vhs&gt;</td>
<td>&lt;russia, soviet&gt;</td>
<td>&lt;zuckerberg, stoll&gt;</td>
<td>&lt;US, US&gt;</td>
</tr>
<tr>
<td>3</td>
<td>&lt;mp3, discman&gt;</td>
<td>&lt;macintosh, digital&gt;</td>
<td>&lt;yeltsin, gorbachev&gt;</td>
<td>&lt;networking, encryption&gt;</td>
<td>&lt;yasukuni, hirohito&gt;</td>
</tr>
<tr>
<td>4</td>
<td>&lt;audio, audio&gt;</td>
<td>&lt;smac, disk&gt;</td>
<td>&lt;usviet, coop&gt;</td>
<td>&lt;coulton, gmel&gt;</td>
<td>&lt;tokyo, tokyo&gt;</td>
</tr>
<tr>
<td>5</td>
<td>&lt;gigabyte, millimeter&gt;</td>
<td>&lt;brion, guber&gt;</td>
<td>&lt;lazovoi, engver&gt;</td>
<td>&lt;sclay, davida&gt;</td>
<td>&lt;korea, korea&gt;</td>
</tr>
<tr>
<td>6</td>
<td>&lt;sony, nintendo&gt;</td>
<td>&lt;gigabyte, millimeter&gt;</td>
<td>&lt;kremlin, republics&gt;</td>
<td>&lt;xanga, cryptography&gt;</td>
<td>&lt;abe, yasuhiro&gt;</td>
</tr>
<tr>
<td>7</td>
<td>&lt;digital, digital&gt;</td>
<td>&lt;simac, vc&gt;</td>
<td>&lt;treaty, anti&gt;</td>
<td>&lt;yahoo, workstations&gt;</td>
<td>&lt;yen, yen&gt;</td>
</tr>
<tr>
<td>8</td>
<td>&lt;nano, rechargeables&gt;</td>
<td>&lt;iphone, dat&gt;</td>
<td>&lt;moscow, moscow&gt;</td>
<td>&lt;boink, gilmore&gt;</td>
<td>&lt;takenaka, nobora&gt;</td>
</tr>
<tr>
<td>9</td>
<td>&lt;portable, portable&gt;</td>
<td>&lt;mp3, cd&gt;</td>
<td>&lt;united, union&gt;</td>
<td>&lt;semel, rotenberg&gt;</td>
<td>&lt;fukui, takeshita&gt;</td>
</tr>
<tr>
<td>10</td>
<td>&lt;music, cassette&gt;</td>
<td>&lt;ipod, video&gt;</td>
<td>&lt;chechnya, russia&gt;</td>
<td>&lt;web, internet&gt;</td>
<td>&lt;shibun, shibun&gt;</td>
</tr>
<tr>
<td>11</td>
<td>&lt;macintosh, rom&gt;</td>
<td>&lt;nano, disks&gt;</td>
<td>&lt;chechen, russian&gt;</td>
<td>&lt;google, computer&gt;</td>
<td>&lt;exports, imports&gt;</td>
</tr>
<tr>
<td>12</td>
<td>&lt;video, video&gt;</td>
<td>&lt;pc, stereo&gt;</td>
<td>&lt;oligarchs, communist&gt;</td>
<td>&lt;mail, mail&gt;</td>
<td>&lt;militarism, anti&gt;</td>
</tr>
<tr>
<td>13</td>
<td>&lt;zune, dat&gt;</td>
<td>&lt;computer, tape&gt;</td>
<td>&lt;khodorkovsky, bakatin&gt;</td>
<td>&lt;oleyourny, menlo&gt;</td>
<td>&lt;trillion, surplus&gt;</td>
</tr>
<tr>
<td>14</td>
<td>&lt;cd, cd&gt;</td>
<td>&lt;music, cassette&gt;</td>
<td>&lt;russians, plotters&gt;</td>
<td>&lt;silicon, electronic&gt;</td>
<td>&lt;whaling, whaling&gt;</td>
</tr>
<tr>
<td>15</td>
<td>&lt;bluetooth, vhs&gt;</td>
<td>&lt;portable, compact&gt;</td>
<td>&lt;gurbaches, kravchuk&gt;</td>
<td>&lt;sek, ip&gt;</td>
<td>&lt;nuclear, fss&gt;</td>
</tr>
<tr>
<td>16</td>
<td>&lt;microsoft, matsushita&gt;</td>
<td>&lt;mini, walkman&gt;</td>
<td>&lt;bush, bush&gt;</td>
<td>&lt;hafi, proteus&gt;</td>
<td>&lt;emperor, emperor&gt;</td>
</tr>
<tr>
<td>17</td>
<td>&lt;download, tape&gt;</td>
<td>&lt;digital, hdtv&gt;</td>
<td>&lt;lehrer, schuster&gt;</td>
<td>&lt;sites, networks&gt;</td>
<td>&lt;ministry, ministry&gt;</td>
</tr>
<tr>
<td>18</td>
<td>&lt;podcasts, disks&gt;</td>
<td>&lt;intel, matsushita&gt;</td>
<td>&lt;nuclear, nuclear&gt;</td>
<td>&lt;users, users&gt;</td>
<td>&lt;trade, trade&gt;</td>
</tr>
<tr>
<td>19</td>
<td>&lt;podcasting, stereo&gt;</td>
<td>&lt;computers, recorders&gt;</td>
<td>&lt;ukraine, ukraine&gt;</td>
<td>&lt;comscore, networked&gt;</td>
<td>&lt;prime, prime&gt;</td>
</tr>
<tr>
<td>20</td>
<td>&lt;electronics, electronics&gt;</td>
<td>&lt;macs, cassettes&gt;</td>
<td>&lt;stalin, reformers&gt;</td>
<td>&lt;murdock, stahlman&gt;</td>
<td>&lt;minister, minister&gt;</td>
</tr>
</tbody>
</table>

REFERENCES


