DEIM Forum 2016 A3-3

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 [12]. 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-formation 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. Kahabua 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 sematic 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 DEFITION

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, $\boldsymbol{Ssim}(e_1,e_2) = \{w_0 \approx \omega_0,\ldots,w_i = \omega_i\}$ where w_i and ω_i exist in e_1 and e_2 respectively; (2) the set of their differences, $\boldsymbol{Sdiff}(e_1,e_2) = \{w_j(e_1),\ldots, \omega_j(e_2)\}$ where $w_j(e_1)$ exists in e_1 but not in e_2, $\omega_j(e_2)$ denotes the opposite.

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

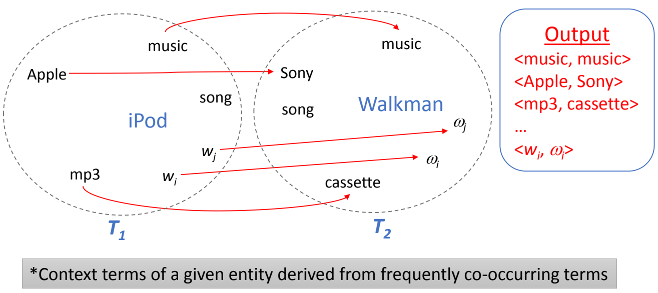

Fig. 1. Conceptual view of pair detection.

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 $m \times p$ 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 $n \times q$ matrix is constructed from the documents in another space (e.g., the documents exist in 1980s), $D(T_2)$ (as shown in Fig. 2).

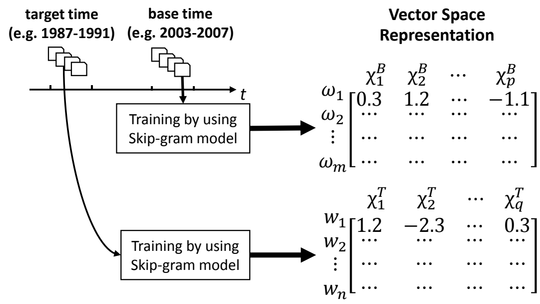

Fig. 2. Creating word vector representations for two spaces.

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 for simplicity).

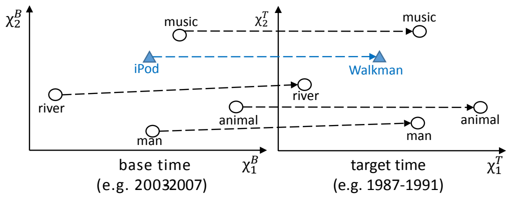

Fig. 3: Conceptual view of the across vector spaces transformation by matching similar relative geometric positions in each space.

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 would 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 $\{(\omega_1, w_1),\ldots,(\omega_k, w_k,)\}$ where ω_i is a anchor in one space and w_i is its counterpart anchor in another space. The transformation matrix $\boldsymbol{M}$ is then found by minimizing the differences between $\boldsymbol{M}\cdot\boldsymbol{\omega_i}$ and $\mathbf{w_i}$ (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 (γ equals to 0.02) with the regularization component added to prevent overfitting.

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

5. SIMILARITY AND DIFFERENCE DETECTION

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

Difference between entities. w_i is a context term of entity e_1, denoted as $w_i(e_1)$ such that it denotes a concept existing in e_1 but not in e_2. $w_i(e_1)$ can explain the difference of e_1 from e_2. Similarly, ω_i is a context term of entity e_2, denoted as $\omega_i(e_2)$ such that it represents a concept existing in e_2 but not in e_1. $\omega_i(e_2)$ can then explain the difference of e_2 from e_1. w_i (or ω_i) should at least satisfy some of the following characteristics: (a) w_i (or ω_i) is highly relevant to e_1 (or e_2); (b) the concept behind w_i (or ω_i) 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.

$$
\begin{aligned}
rel(w,e) &= P(w \mid e) \cdot P(e \mid w) \\
&= \frac{P(w,e)}{P(e)} \times \frac{P(w,e)}{P(w)} \propto \frac{P(w,e)^2}{P(w)} \\
&= \left(\frac{n_{w,e}}{\max\limits_{t \in d(e)}(n_{t,e})}\right)^2 \times \frac{\max\limits_{t \in D}(n_t)}{n_w}
\end{aligned} \tag{2}
$$

Note that Eq. 2 computes the relevance between each context term of a given input entity, e.g., $rel(w_i,e_1)$ and $rel(\omega_i,e_2)$ 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, $rel(<w_i,\omega_i>)$, can be computed as Eq. 3.

$$
\begin{aligned}
rel(< w_i, \omega_i >) &= rel(< w_i, e_1 >, < \omega_i, e_2 >) \\
&= rel(w_i, e_1) \cdot rel(\omega_i, e_2)
\end{aligned} \tag{3}
$$

Intra-Similarity. This feature measures the similarity between a pair of context terms $<w_i,\omega_i>$ where w_i is the context term of entity e_1 and ω_i is the context term of entity e_2. As discussed in Sec. 4, since w_i and ω_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., $\boldsymbol{w}_i$) from one vector space (e.g., T_1) to another space (e.g., T_2). We then compare the transformed representation (e.g., $\boldsymbol{M}^T \cdot \boldsymbol{w}_i$) with the representation of the term in T_2 (e.g., $\boldsymbol{\omega}_i$) by cosine similarity. The intra-similarity of $<w_i,\omega_i>$ is computed by Eq. 4.

$$
simIntra(< w_i, \omega_i >) = \cos(M \cdot w_i, \omega_i) \tag{4}
$$

Relational-Similarity. Besides the intra-similarity discussed above which measures the semantic similarity between w_i and ω_i, the relational similarity estimates if the relation between w_i and e_1 is corresponds to the relation between ω_i and e_2. To represent the relation between w_i and e_1, we take the difference of their vector representations, $\boldsymbol{w}_i - \boldsymbol{e}_1$. The relational similarity is defined as below.

$$
\begin{aligned}
& simR(< w_i, e_1 >, < \omega_i, e_2 >) \\
&= \cos(M \cdot (w_i - e_1), (\omega_i - e_2))
\end{aligned} \tag{5}
$$

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 ($\boldsymbol{S_{sim}}$) and a set of evidences denoting the difference ($\boldsymbol{S_{diff}}$) between two entities.

We define a good set $\boldsymbol{S_{sim}}$ (or $\boldsymbol{S_{diff}}$) 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 $\boldsymbol{S_{sim}}$ as the pair of terms $<w_i,\omega_i>$ which should have high in relevance score, high in intra similarity and high in relational similarity (see Eqs. 6-7).

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

$$
sim(< w_i, \omega_i >) = simIntra(< w_i, \omega_i >) \cdot simR(< w_i, e_1 >, < \omega_i, e_2 >) \tag{7}
$$

On the other hand, the quality of the item in $\boldsymbol{S_{diff}}$ 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 (ω_i) in e_2 and low in the relation similarity (see Eq. 8).

$$
qua_{diff}(w_i(e_1)) = rel(w_i, e_1)^{\alpha} \cdot (1 - \max_{\omega_j \in C} sim(< w_i, \omega_j >)^{1-\alpha} \tag{8}
$$

where, C is the set of context terms of e_2. $sim(<w_i,\omega_i>)$ 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.

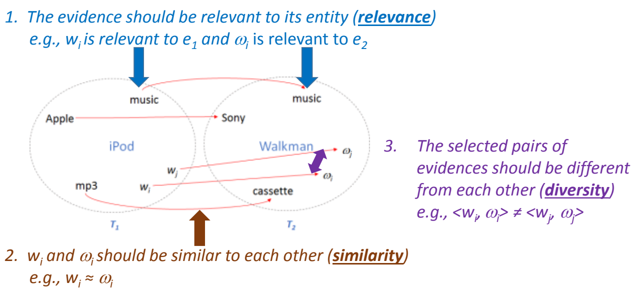

Fig. 4: Conceptual view of the optimization process.

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.

$$
L = \arg\max_{A_i \in U-S} \{\lambda \cdot qua(A_i) - (1-\lambda) \max_{A_v \in S}(simInter(A_v, A_i))\} \tag{9}
$$

$$
\begin{cases}
simInter(A_v, A_i) = \cos(< w_v, \omega_v >, < w_i, \omega_i >) & \\
\qquad = \cos(w_v, w_i) \cdot \cos(\omega_v, \omega_i) & \text{for } S_{sim}, \\
simInter(A_v, A_i) = \cos(w_v, w_i) & \text{for } S_{diff},
\end{cases} \tag{10}
$$

L is the size of the returned set. S denotes the selected set and A_v 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_{sim} when replacing $qua(A_i)$ by Eq. 6 or to optimize the evidences S_{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 before[1] 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.

[*The evaluations over different metrics and the performance comparison between proposed methods and the baseline methods will be included later.*]

8. CONCLUSTION 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 e_1 is the entity in the time [2002,2007] and e_2 is in [1987,1991]. In the experiments, the size of the returned evidence set equals 20.

(e_1,e_2)	**(ipod,walkman)**	**(apple,sony)**	**(putin,yeltsin)**	**(facebook,usenet)**	**(japan,japan)**
1	<apple,sony>	<ipod,vhs>	<russian,mikhail>	<myspace,unix>	<koizumi,kaifu>
2	<itunes,cassettes>	<itunes,audio>	<russia,soviet>	<zuckerberg,stoll>	<US,US>
3	<mp3,discman>	<macintosh,digital>	<yeltsin,gorbachev>	<networking,encryption>	<yasukuni,hirohito>
4	<audio,audio>	<mac,disk>	<soviet,coup>	<coulton,gemmel>	<tokyo,tokyo>
5	<gigabyte,milimeter>	<brion,guber>	<lugovoi,engver>	<cloyd,davida>	<korea,korea>
6	<sony,nintendo>	<gigabyte,millimeter>	<kremlin,republics>	<xanga,cryptography>	<abe,yasuhiro>
7	<digital,digital>	<imac,vcr>	<treaty,anti>	<yahoo,workstations>	<yen,yen>
8	<nano,rechargeables>	<iphone,dat>	<moscow,moscow>	<boink,gilmore>	<takenaka,noboru>
9	<portable,portable>	<mp3,cd>	<united,union>	<semel,rotenberg>	<fukui,takeshita>
10	<music,cassette>	<ipods,video>	<chechnya,russia>	<web,internet>	<shimbun,shimbun>
11	<macintosh,rom>	<nano,disks>	<chechen,russian>	<google,computer>	<exports,imports>
12	<video,video>	<pc,stereo>	<oligarchs,communist>	<mail,mail>	<militarism,anti>
13	<zune,dat>	<computer,tape>	<khodorkovsky,bakatin>	<oleyourryk,menlo>	<trillion,surplus>
14	<cd,cd>	<music,cassette>	<russians,plotters>	<silicon,electronic>	<whaling,whaling>
15	<bluetooth,vhs>	<portable,compact>	<gorbachev,kravchuk>	<ck,ipc>	<nuclear,fsx>
16	<microsoft,matsushita>	<mini,walkman>	<bush,bush>	<hali,proteus>	<emperor,emperor>
17	<download,tape>	<digital,hdtv>	<lehrer,schuster>	<sites,networks>	<ministry,ministry>
18	<podcasts,disks>	<intel,matsushita>	<nuclear,nuclear>	<users,users>	<trade,trade>
19	<podcasting,stereo>	<computers,recorders>	<ukraine,ukraine>	<comScore,networked>	<prime,prime>
20	<electronics,electronics>	<macs,cassettes>	<stalin,reformers>	<Murdoch,stahlman>	<minister,minister>

REFERENCES

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

[2] O. Alonso, M. Gertz, R. Baeza-Yates. Clustering and exploring search results using timeline constructions. In Proc. of CIKM, pp. 97-106, 2009.

[3] O. Alonso, K. Shiells. Timelines as summaries of popular scheduled events. In Proc. of WWW, pp. 1037-1044, 2013.

[4] R. Baeza-Yates. Searching the future. In SIGIR Workshop MF/IR, 2005.

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

[6] K. Berberich, M. Vazirgiannis, G. Weikum. Time-aware authority ranking. Internet Mathematics, 2(3): 301-332, 2005.

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

[8] E. Borovikov. A survey of modern optical character recognition techniques. arXiv preprint arXiv:1412.4183, 2014.

[9] J. Blitzer, R. McDonald, and F. Pereira. Domain adaption with structural correspondence learning. In Proc. of EMNLP, pp. 120-128, 2006.

[10] P. Cai, W. Gao, A. Zhou et al. Relevant knowledge helps in choosing right teacher: active query selection for ranking adaptation. In Proc. of SIGIR, pp. 115-124, 2001.

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

[12] R. Campos, G. Dias, A. M. Jorge and A. Jatowt. Survey of temporal information retrieval and related applications. ACM Computing Surveys (CSUR), 47(2), 15, 2014.

[13] W. Dakka, L. Gravano, G. Ipeirotis. Answering general time-sensitive queries. IEEE TKDE, 24(2): 220-235, 2012.

[14] S. Deerwester et al., Improving Information Retrieval with Latent Semantic Indexing, In Proc. of the 51st

Annual Meeting of the American Society for Information Science, 25, pp. 36–40, 1988.

[15] L. Elsas, T. Dumais. Leveraging temporal dynamics of document content in relevance ranking. In Proc. of WSDM, pp.1-10, 2010.

[16] W. Gao, P. Cai, K.F. Wong et al. Learning to rank only using training data from related domain. In Proc. of SIGIR, pp. 162-169, 2010.

[17] Z. Harris, Distributional Structure, Word, 10(23):146–162, 1954.

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

[19] A. Jatowt A, C. M. Au Yeung. Extracting collective expectations about the future from large text collections. In Proc. of CIKM'11, pp. 1259-1264.

[20] A. Jatowt and K. Duh. A framework for analyzing semantic change of words across time. In Proc. of JCDL, pp. 229-238, 2014.

[21] A. Jatowt, K. Kanazawa, S. Oyama, et al. Supporting analysis of future-related information in news archives and the web. In Proc. of JCDL, pp. 115-124, 2009.

[22] A. Kalurachchi, A. S. Varde, S. Bedathur, et al., Incorporating Terminology Evolution for Query Translation in Text Retrieval with Association Rules, In Proc. of CIKM, pp. 1789-1792, 2010.

[23] N. Kanhabua, K. Nørvåg. Determining time of queries for re-ranking search results, Research and Advanced Technology for Digital Libraries. Springer Berlin Heidelberg, pp. 261-272, 2010.

[24] N. Kanhabua, K. Nørvåg, Exploiting Time-based Synonyms in Searching Document Archives, In Proc. of JCDL, pp. 79-88, 2010.

[25] N. Kanhabua, K. Nørvåg. A comparison of time-aware ranking methods. In Proc. of SIGIR, pp. 1257-1258, 2011.

[26] M. P. Kato, H. Ohshima and K. Tanaka. Content-based Retrieval for Heterogeneous Domains: Domain Adaption by Relative Aggregation Points. In Proc. SIGIR, pp. 811-820, 2012.

[27] Y. Kim, Y-I. Chiu, K. Hanaki, D. Hegde and S. Petrov. Temporal Analysis of Language through Neural Language Models. In Proc. of ACL Workshop, pp. 61-65, 2014.

[28] V. Kulkarni, R. Al-Rfou, B. Perozzi, and S. Skiena. Statistically Significant Detection of Linguistic Change. In Proc. of WWW'15, pp. 625-635.

[29] W. Labov. Principles of Linguistic Change, Wiley-Blackwell, 2010.

[30] X. Li, W. B. Croft. Time-based language models, In Proc. of CIKM, pp. 469-475, 2003.

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

[32] X. Ling, W. Dai, G. R. Xue, Q. Yang and Y. Yu. Spectral domain-transfer learning. In Proc. of SIGKDD, pp. 488-496, 2008.

[33] D. Metzler, R. Jones, F. Peng, et al. Improving search relevance for implicitly temporal queries. In Proc. of SIGIR, pp. 700-701, 2009.

[34] R. Mihalcea, and V. Nastase, Word Epoch Disambiguation: Finding How Words Change Over Time. In Proc. of ACL, pp. 259-263, 2012.

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

[36] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representation of Phrases and Their Compositionality. In Proc. of NIPS, pp. 3111-3119, 2013.

[37] P. Norvig. Natural language corpus data. Beautiful Data, pp. 219-242, 2009. http://norvig.com/spell-correct.html

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

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

[40] M. Pargel, Q. D. Atkinson and A. Meade. Frequency of word-use predicts rates of lexical evolution throughout Indo-European history. Nature, 449, 717-720, 2007.

[41] M. W. C. Reynaert. Character confusion versus focus word-based correction of spelling and OCR variants in corpora. International Journal on Document Analysis and Recognition, 14(2): 173-187, 2011.

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

[43] E. Sandhaus. The New York Times Annotated Corpus Overview. The New York Times Company, Research & Develop., pp. 1-22, 2008.

[44] M. Steinbach, G. Karypis, V. Kumar. A comparison of document clustering techniques. In Proc. of KDD workshop, 400(1): 525-526, 2000.

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

[46] N. K. Tran, A. Ceroni, Kanhabua N, et al. Back to the past: Supporting interpretations of forgotten stories by time-aware re-contextualization. In Proc. of WSDM, pp. 339-348, 2015.

[47] H. Wang, H. Huang, F. Nie, and C. Ding. Cross-language web page classification via dual knowledge transfer using nonnegative matrix tri-factorization. In Proc. of SIGIR, pp. 933-942, 2011.

[48] B. Wang, J. Tang, W. Fan, S. Chen, Z. Yang and Y. Liu. Heterogeneous cross domain ranking in latent space. In Proc. of CIKM'09, pp. 987-996.

[49] Y. Zhang, A. Jatowt, S. S. Bhowmick and K. Tanaka. Omnia Mutantur, Nihil Interit: Connecting Past with Present by Finding Corresponding Terms across Time. In Proc. of ACL, pp. 645-655, 2015.