DEIM Forum 2017 G8-3

Cluster-biased Transformation of Texts across Heterogeneous Domains

Yating ZHANG[†], Adam JATOWT[†], and Katsumi TANAKA[†]

† Graduate School of Informatics, Kyoto University
Yoshida-Honmachi, Sakyo-ku, Kyoto, 606-8501 Japan
E-mail: †{zhang,adam,tanaka}@dl.kuis.kyoto-u.ac.jp

Abstract In the recent years we have witnessed a rapid increase of diverse text content generated on the Web or digitized in archives. It is then more difficult for users to perform search and comparison within large volumes of text data, due to the terminology gap between the domain of user's knowledge and the search domain. For example, users often struggle when searching in unfamiliar domains if they do not know correct keywords (e.g., searching for information related to a specific time or a foreign country) or when comparing documents generated by different sources. In this paper, we approach the problem of terminology gap by providing a general framework to bridge different domains and, by this, to facilitate search and comparison as if they would be carried in user's familiar domains. We first propose a cluster-biased transformation technique which allows mapping words across different domains and we then adapt this method to enable mapping entire documents (e.g., set of words, relational graphs) across domains. The proposed methods are unsupervised and can be applied to any scenario. We test the performance of our approaches on documents from geographically as well as temporally different domains.

Key words text transformation, cluster-biased, heterogeneous domains

1. INTRODUCTION

In the recent years we have witnessed a rapid increase of text content generated on the Web or digitized in archives. The current keyword-based search engine thus has limited performance in supporting users to perform search and comparison within large volumes of text data, due to the terminology gap between user's knowledge domain and the search domain. For example, when searching in unfamiliar domains, such as searching within past collections or collections of documents related to different countries, users have often difficulty to recall correct keywords to search with. The vocabulary mismatch resulting from heterogeneous document collections also hinders the comparison of documents published in different periods due to the terminology change across time or in diverse locations where cultures/objects/entities are quite different. The search and comparison of texts across heterogeneous domains is the research focus of this paper.

To bridge the gap between a user's familiar domain and the unfamiliar search domain for enabling easy text search and comparison, we propose a cluster-biased transformation mechanism which automatically transforms the text from one domain to the other as well as establishing the across-domain similarity measures, thus allowing users to search by analogical objects or compare different documents. In this paper, we focus on the process of constructing the connections of text across different collections (or text transformation process). Once the transformation is established, different retrieval requests/applications can be accessed/applied to. For instance, in the research problem of searching by analogical objects as we mentioned before, users can search by "iPod in 1980s" (replacing the correct keyword "Walkman") or "Japanese NASA" (instead of the keyword "JAXA"), when the keyword "Walkman" or "JAXA" is unknown to them. The comparison between long texts (e.g., sentences or documents) can be also conducted by applying across-domain similarity measures introduced in this paper.

Transformation mechanism for text does not only help to solve the cold-start problem in search across domains by formulating queries in the form of analogy, but it also enables the text comparison (e.g., word, sentence, document comparison) in different collections. However, transformation of text in heterogeneous domains is not trivial. The main challenge lies in the vocabulary gap across collections due to the time change, culture differences etc, suggesting that the direct context comparison will not work. To solve this issue, we apply distributed word embedding technique [6], [7] in order to first set the vocabularies of each domain into their own semantic spaces separately. Then, we construct similarity measure of words across domains by mapping the vocabularies across vector spaces.

To design the mapping function as mentioned above, we first propose to utilize automatically derived training seeds to construct one transformation matrix for aligning two vector spaces by assuming that all the vocabularies in one space follow the same rule (mapping function) in transformation. In this paper we also introduce an advanced approach named cluster-biased term transformation to relax the above assumption by training multiple transformation matrices biased on specific semantic clusters across vector spaces. This advanced approach 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 further across-domain search or document comparisons, we introduce several retrieval models and similarity metrics based on the above mentioned transformation mechanisms.

To sum up, our contributions in this work are as follows:

（1） We propose an efficient and effective framework to transform text across heterogeneous domains, which provides fundamental technique to conduct similarity comparison across vector spaces. For more precise mapping, we then introduce an advanced cluster-biased text transformation mechanism to train specific mapping function for each semantic cluster.

（2） To test the performance of the proposed transformation techniques, we provide several retrieval models for across-domain search scenario and text comparison.

（3） We evaluate the proposed approaches on the unstructured text in both temporal collections and spatial datasets, which prove the effectiveness of our approach.

The reminder of this paper is structured as follows. We formally describe the research problem in the next section. We then discuss the general term transformation technique for aligning two vector spaces in Section 3. Section 4 explains our advanced approach of cluster-biased term transformation. Next, we describe the experimental setup and give the evaluation results in Sections 5 and 6, respectively. The next section contains the discussion. We conclude the paper and outline the future work in the last section.

2. PROBLEM STATEMENT

In this section, we formally define the problem of the term transformation across heterogeneous domains.

We set two spaces: a base domain (user's knowledge domain) $S^b = \{w_1^b, w_2^b, ..., w_m^b\}$ ($w_i^b \in$ Vocabulary of S^b) from which the query (or document) is selected, and a target domain (user's search domain) $S^t = \{w_1^t, w_2^t, ..., w_n^t\}$ ($w_i^t \in$ Vocabulary of S^t) where the answer is to be retrieved from.

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

Across-Domain Similar Object is defined as an object w^t (e.g., `JAXA`) which is contextually similar to the queried object w^b (e.g., *NASA*) (in the scenarios of searching across locations (Japan vs. USA)). Note that the context between w^t and w^b is not required to be literally same. The literal form could be different as long as their meanings are similar.

Across-Domain Similar Document is defined as a document $d^t = \{(w_1^t, f_1^t), (w_2^t, f_2^t), ..., (w_m^t, f_m^t)\}$ (where f_i^t is the frequency of term w_i in d^t) which is semantically similar to queried document $d^b = \{(w_1^b, f_1^b), (w_2^b, f_2^b), ..., (w_m^b, f_m^b)\}$.

3. GLOBAL TERM TRANSFORMATION

Our goal is to compare terms related to disjoint geographical areas and to find matching term pairs (e.g., `NASA` and `JAXA`). For this, we propose constructing a mapping function between the base space and the target space. This process is query independent and can be done offline. While establishing the mapping function would necessarily require a supervised method, we assume in this work an unsupervised approach. This is because it is infeasible to provide sufficient number of training pairs of terms (such as the examples listed above) for any possible combination of two different countries or other geographic regions. We then resort to automatically finding training pairs. One way to generate such term pairs could be based on the equality of term literal forms. However, we cannot always assume a direct semantic correspondence between the same term in two different spaces. Even if the same term appears in two different spaces, there is no assurance that it indeed denotes identical concept. For example, `sushi` in Japan is regarded as the typical or local food, however in another country, such as USA, though `sushi` also exists, the position/role behind it is rather different (e.g., `sushi` is regarded as foreign and relatively luxury food in USA). This phenomena can be interpreted as the meaning shift across spaces. On the other hand, sometimes literally different terms in different spatial areas may represent the same or very similar concept, such as `haiku` in Japan and `poetry` in USA.

3.1 Word Embedding

For capturing word semantics we use word embedding techniques. Distributed representation of words by using neural networks was originally proposed in [9]. Mikolov *et al.* [6], [7] 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.

3.2 Transformation based on Anchor Mapping

Our goal is to compare words in the base space and words in the target space in order to find their counterparts. Since, as mentioned before, it is impossible to directly compare words in two different semantic vector spaces (the features/dimensions in both spaces have no direct correspondence due to separate training processes), we train a transformation matrix to build the connection between different 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 the 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 established the correspondence between the anchor terms in the two semantic spaces, we can automatically map any terms relative to these anchors. In this work, we propose to use *Shared Frequent Concepts* (SFC) as anchors to build the transformation. To construct the transformation, however, manually preparing large enough 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 extracting SFC as anchor pairs. Specifically, we select terms that (a) are general in their meaning and (b) have high frequency (e.g., mountain, river, lake, president) in both the base and the target spaces. The intuition behind this idea is that general terms that are also frequent in both spaces are more likely to have stable meaning and be also co-occurring with many other terms.

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 is an anchor in one space (e.g., Japan) and x_i^t is its counterpart, that is, the same anchor in the other space (e.g., USA). The transformation matrix $\mathbf{M}$ is found 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 ($\gamma = .02$) with regularization component used for preventing overfitting:

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

u denotes here the size of anchor term set which contains, in our implementation, the top 5% frequent concepts (over 10,000 terms) in the intersection of vocabularies of the two corpora.

3.3 Retrieval Model for Global Term Transformation

After obtaining the transformation matrix $\mathbf{M}$, we can then compute the similarity of a query, q, in the base space with any term v in the target space by multiplying the query's vector representation with the transformation matrix $\mathbf{M}$, and then by calculating the cosine similarity between such transformed vector and the vector v.

$$S_{sim}(\mathbf{q}, \mathbf{v}) = cos(\mathbf{Mq}, \mathbf{v}) \qquad (2)$$

4. CLUSTER-BIASED TERM TRANSFORMATION

In Sec. 3, we explained the way to construct a single transformation matrix for aligning two vector spaces by assuming that all the vocabularies in one space follow the same mapping function, which is obviously a simplified approach. To relax the above assumption, we propose an advanced approach, called cluster-biased term transformation, to train multiple transformation matrices biased on specific semantic clusters across vector spaces. The motivation behind this approach lies in the notion that a single transformation matrix is too coarse and general to serve well for any possible queries. We believe that the combination of "local" approaches designed for semantic subspaces should work better. In the new approach we consider that the characteristics of the word embedding spaces are such that in each vector space, the semantically similar words are located close to each other. We then propose a clustering based 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 of each space by hierarchical clustering and then establish the mapping function across clusters in two domains.

4.1 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 single-linkage criterion (implemented by SLINK [11] algorithm(注1)) to determine the distance between clusters when doing the merging, that is the minimum distance between elements of each cluster where the distance of words are measured by the inverse of cosine similarity between their word embeddings. HAC process is done separately for each vector space. After the HAC is completed, we can obtain a hierarchical clusters. Then each word in a vector space belongs to a hierarchical path of clusters from the leaf cluster (the word itself) to the root cluster

(注1): Considering the lower complexity of the SLINK algorithm: O(n^2) in time complexity and O(n) in space complexity which contains all the words.

4.2 Term Transformation biased on Semantic Clusters

As discussed in the beginning of Sec. 4, we aim at training transformation matrices for different semantic clusters. Different from training of global term transformation which give equal weights to all the seeds (see Eq. 1), in cluster-biased approach, our idea is that for each semantic cluster, we bias on the seeds which highly related to this cluster by manipulating the weights (λ) on the seeds when training the transformation matrix (see Eq. 3).

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

The weight $\lambda_{i,k}$ of seed x_i biasing on cluster C_k^b is computed by Eq. 4.

$$\lambda_{i,k} = e^{-(L_{C_k^b} - L_{C_j^b})} \qquad (4)$$

where $L_{C_k^b}$ (or $L_{C_j^b}$) denotes the length (or 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 word x_i and meanwhile (3) it is the nearest cluster to C_k^b.

Toy Example. We show a toy example for better explaining how to calculate the weights of the seeds for a specific cluster. In our toy example, the hierarchy tree of the words in base space is shown in Fig. 1 and we specifically compute the seed weights for semantic cluster $C_{k=4}$. Suppose the seeds for training transformation matrix are $\{x_1, x_2, x_3\}$, then Table 1 computes the weights of these seeds step by step following Eq. 4.

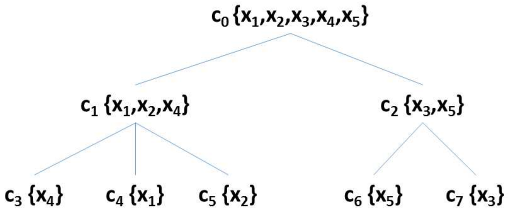

Figure 1 Toy example of hierarchy tree with seeds denoted as x_i

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

seed	path	C_j	$L_{C_{k=4}} - L_{C_j}$	$\lambda_{i,k=4}$
x_1	$\{C_4, C_1, C_0\}$	C_4	2-2=0	1
x_2	$\{C_5, C_1, C_0\}$	C_1	2-1=1	e^{-1}
x_3	$\{C_7, C_2, C_0\}$	C_0	2-0=2	e^{-2}

After computing the seed weights, they can be normalized over all the seeds and input to Eq. 3 to obtain a term transformation matrix biased on cluster $C_{k=4}$.

Note that when $k = 0$ (root cluster), the seed weights will be all equal, which is as the same as the approach of global term transformation introduce in Sec. 3. Therefore, global term transformation (Eq. 1) can be regarded as a special case of cluster-biased term transformation (Eq. 3).

4.3 Retrieval Model for Cluster-biased Term Transformation

As discussed above, for each semantic cluster, we train a transformation matrix in the way described in Eq. 3. However each word has multiple memberships over different clusters from the word itself to the root cluster and each transformation process will obtain a list of candidate results. Therefore, in this section we propose to combine the results from different transformations by aggregating the transformed vector of the query (see Eq. 5).

$$S_{sim}(\mathbf{q}, \mathbf{v}) = \frac{1}{K} \sum_{k=1}^{K} cos(\mathbf{M_k q}, \mathbf{v}) \qquad (5)$$

5. EXPERIMENTAL SETTINGS

5.1 Datasets

Across-time search. For the experiments we use the New York Times Annotated Corpus [10]. This dataset contains over 1.8 million newspaper articles published between 1987 and 2007. We first divide it into five parts according to article publication time: [1987-1991], [1992-1996], [1997-2001] and [2002-2007]. Each time period contains then around half million articles. We next train the model of distributed vector representation separately for each time period. The vocabulary size of the entire corpus is 360k, while the vocabulary size of each time period is around 300k.

5.2 Test sets

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

In total, we have 225 pairs of query term and its counterpart for the task of mapping [2002-2007] with [1987, 1991] and 100 test pairs for the task of mapping [2002-2007] with [1992, 1996].

5.3 Tested Methods

5.3.1 Baselines

We prepare five baselines as follows:

(1) **Word embedding model without transformation (NT)**: **NT** uses distributional representation for capturing

Table 2 [2002,2007]→[1992,1996]

	MRR (Impr.%)	P@1 (%)	@5 (%)	@10 (%)	@20 (%)	@50 (%)
HT	0.213 (+16.9)	8.7	32.6	42.4	58.7	77.2
GT	0.182	7.6	27.2	33.7	50.0	63.0
NT	0.119 (-34.6)	3.8	17.0	23.8	30.1	41.7

Table 3 [2002,2007]→[1987,1991]

	MRR (Impr.%)	P@1 (%)	@5 (%)	@10 (%)	@20 (%)	@50 (%)
HT	0.216* (+21.0)	11.1	**32.4**	43.6	**63.1**	**78.7**
GT	0.179	10.2	23.6	35.1	49.3	69.8
NT	0.129 (-27.8)	4.5	17.9	21.4	28.4	35.8

word semantics same as the proposed methods do. Instead of training the document collections from two periods separately, it however trains a joint vector space by mixing all the documents. We can then evaluate the necessity of the transformation by testing this method in comparison to the proposed methods.

(2) **Global Transformation** (**GT**) maps the terms in one vector space to the other by training a global transformation matrix as described in Sec. 3. We set **GT** as control baseline to verify the effectiveness of our proposed transformation technique based on dual hierarchical structures as described in Sec. 4.

5.3.2 Proposed Methods

HT conducts term transformation at each cluster on the path of query to the root following the base hierarchical tree (see Sec. 4.

6. EXPERIMENTAL RESULTS

As shown in Tab. 2 and 3, the method **HT** statistically significantly ($p < .1$) outperforms the best baseline **GT**

6.0.1 Necessity of Transformation

The next observation is that method **NT** achieves relatively low performance. **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.

6.0.2 Improvement by Leveraging Hierarchical Cluster Information

In Sec. 4, we have proposed a cluster-biased transformation techqnieu for boosting the search performance. As seen in Tab. 2 and 3, we can observe an increase of performance by incorporating the information about hierarchical clustering information when performing transformation. **HT** has on average 19% better performance than **GT**

7. RELATED WORK

Several researchers [1], [4], [5], [8] have approached domain adaptation task. Blitzer *et al.* [1] proposed a Structural Correspondence Learning (SCL) to identify correspondences among features from different domains by modeling their correlations with pivot features. The method was proved to perform well in a discriminative framework, such as in the task of PoS-tagging. Similarly, Kato *et al.* [4], [5] proposed to utilize Relative Aggregation Point (RAP) such as average price, maximum/minimum cost, restaurant categories etc. in different domains as features to detect a corresponding restaurant in another city. Both of these approaches were done in a discriminative learning manner where a conditional probability of the instances in a domain was estimated and classified into a certain class. However, these approaches only work for the data where the instances are already classified or the distributions of the instances over categories are known in a domain. For many datasets, such as news archives, online reviews, encyclopedias where the entities are unstructured or the entities are not represented by any fixed attributes, one needs to leverage other information to solve the domain adaptation problem. Unlike these researches, we propose a general framework by only leveraging the semantics of terms and their relative positions in each semantic space to perform transformation. Our methods can be applied to any orthogonal raw-text datasets while the query can be any term (e.g., city, person, object, culture).

Analogical relation detection [2], [12], [13] is to some extent related to our work. Structure Mapping Engine (SME) [2] was the original implementation of Structure Mapping Theory (SMT) [3] that explains how humans reason with analogy. Later, Turney proposed Latent Relational Mapping Engine (LRME) [13] that extracts lexical patterns in which words co-occur to measure relational similarity. These approaches are always based on a single dataset. This means that in such a case contextual information specific to particular country is lost as also shown in our experiments. Moreover none of the previous works specifically focused on spatial analogy task. Finally, different types of models have been proposed for the task of proportional analogy [13]. However, their objective was to extract an object that can fit into equation $a : b :: c : d$ when one of the four constituents is missing.

8. CONCLUSION

Nowadays, users often search for information related to distant and unknown places. To decrease the problem stemming from the vocabulary gap we propose query suggestion mechanism based on automatic transformation of concepts from one time area to another. The problem is not trivial due to diverse contexts of semantically similar terms within different time periods as demonstrated by poor performance of approaches relying on a joint dataset. We introduce several unsupervised methods for mapping terms from different time periods. An important characteristics of our approach is that it works on raw text collections without the need for utilizing knowledge bases or any supervision.

References

[1] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proc. of EMNLP*, pages 120–128, 2006.

[2] B. Falkenhainer, K. D. Forbus, and D. Gentner. The structure-mapping engine: Algorithm and examples. *Artificial intelligence*, 41(1):1–63, 1989.

[3] D. Gentner. Structure-mapping: A theoretical framework for analogy*. *Cognitive science*, 7(2):155–170, 1983.

[4] M. P. Kato, H. Ohshima, S. Oyama, and K. Tanaka. Search as if you were in your home town: Geographic search by regional context and dynamic feature-space selection. In *Proc. of CIKM*, pages 1541–1544, 2010.

[5] M. P. Kato, H. Ohshima, and K. Tanaka. Content-based retrieval for heterogeneous domains: Domain adaptation by relative aggregation points. In *Proc. of SIGIR*, pages 811–820, 2012.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *Proc. of ICLR Workshop*, 2013.

[7] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representation of phrases and their compositionality. In *Proc. of NIPS*, pages 3111–3119, 2013.

[8] S. Pan and Q. Yang. A survey on transfer learning. *IEEE TKDE*, 22(10):1345–1359, 2010.

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

[10] E. Sandhaus. The new york times annotated corpus overview. *The New York Times Company, Research & Develop.*, pages 1–22, 2008.

[11] R. Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The computer journal*, 16(1):30–34, 1973.

[12] P. D. Turney. Expressing implicit semantic relations without supervision. *CoRR*, 2006.

[13] P. D. Turney. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, pages 615–655, 2008.