Learning Discriminative Projections for Text Similarity Measures

Wen-tau Yih **Kristina Toutanova** **John C. Platt** **Christopher Meek**
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
{scottyih,kristout,jplatt,meek}@microsoft.com

Abstract

Traditional text similarity measures consider each term similar only to itself and do not model semantic relatedness of terms. We propose a novel discriminative training method that projects the raw term vectors into a common, low-dimensional vector space. Our approach operates by finding the optimal matrix to minimize the loss of the pre-selected similarity function (e.g., cosine) of the projected vectors, and is able to efficiently handle a large number of training examples in the high-dimensional space. Evaluated on two very different tasks, cross-lingual document retrieval and ad relevance measure, our method not only outperforms existing state-of-the-art approaches, but also achieves high accuracy at low dimensions and is thus more efficient.

1 Introduction

Measures of text similarity have many applications and have been studied extensively in both the NLP and IR communities. For example, a combination of corpus and knowledge based methods have been invented for judging word similarity (Lin, 1998; Agirre et al., 2009). Similarity derived from a large-scale Web corpus has been used for automatically extending lists of typed entities (Vyas and Pantel, 2009). Judging the degree of similarity between documents is also fundamental to classical IR problems such as document retrieval (Manning et al., 2008). In all these applications, the vector-based similarity method is the most widely used. Term vectors are first constructed to represent the original text objects, where each term is associated with a weight indicating its importance. A pre-selected function operating on these vectors, such as cosine, is used to output the final similarity score. This approach has not only proved to be effective, but is also efficient. For instance, only the term vectors rather than the raw data need to be stored. A pruned inverse index can be built to support fast similarity search.

However, the main weakness of this term-vector representation is that different but semantically related terms are not matched and cannot influence the final similarity score. As an illustrative example, suppose the two compared term-vectors are: {*purchase*:0.4, *used*:0.3, *automobile*:0.2} and {*buy*:0.3, *pre-owned*: 0.5, *car*: 0.4}. Even though the two vectors represent very similar concepts, their similarity score will be 0, for functions like cosine, overlap or Jaccard. Such an issue is more severe in cross-lingual settings. Because language vocabularies typically have little overlap, term-vector representations are completely inapplicable to measuring similarity between documents in different languages. The general strategy to handle this problem is to map the raw representation to a common *concept* space, where extensive approaches have been proposed. Existing methods roughly fall into three categories. Generative topic models like Latent Dirichlet Allocation (LDA) (Blei et al., 2003) assume that the terms are sampled by probability distributions governed by hidden topics. Linear projection methods like Latent Semantic Analysis (LSA) (Deerwester et al., 1990) learn a projection matrix and map the original term-vectors to the dense low-dimensional space. Finally, metric learning approaches for high-dimensional spaces have

Proceedings of the Fifteenth Conference on Computational Natural Language Learning, pages 247–256,
Portland, Oregon, USA, 23–24 June 2011.

also been proposed (Davis and Dhillon, 2008).

In this paper, we propose a new projection learning framework, Similarity Learning via Siamese Neural Network (S2Net), to discriminatively learn the concept vector representations of input text objects. Following the general Siamese neural network architecture (Bromley et al., 1993), our approach trains two identical networks concurrently. The input layer corresponds to the original term vector and the output layer is the projected concept vector. Model parameters (i.e., the weights on the edges) are equivalently the projection matrix. Given pairs of raw term vectors and their labels (e.g., similar or not), the model is trained by minimizing the loss of the similarity scores of the output vectors. S2Net is closely related to the linear projection and metric learning approaches, but enjoys additional advantages over existing methods. While its model form is identical to that of LSA, CCA and OPCA, its objective function can be easily designed to match the true evaluation metric of interest for the target task, which leads to better performance. Compared to existing high-dimensional metric learning methods, S2Net can learn from a much larger number of labeled examples. These two properties are crucial in helping S2Net outperform existing methods. For retrieving comparable cross-lingual documents, S2Net achieves higher accuracy than the best approach (OPCA) at a much lower dimension of the concept space (500 vs. 2,000). In a monolingual setting, where the task is to judge the relevance of an ad landing page to a query, S2Net also has the best performance when compared to a number of approaches, including the raw TFIDF cosine baseline.

In the rest of the paper, we first survey some existing work in Sec. 2, with an emphasis on approaches included in our experimental comparison. We present our method in Sec. 3 and report on an extensive experimental study in Sec. 4. Other related work is discussed in Sec. 5 and finally Sec. 6 concludes the paper.

2 Previous Work

In this section, we briefly review existing approaches for mapping high-dimensional term-vectors to a low-dimensional concept space.

2.1 Generative Topic Models

Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999) assumes that each document has a document-specific distribution θ over some finite number K of topics, where each token in a document is independently generated by first selecting a topic z from a multinomial distribution $\textsc{Multi}(\theta)$, and then sampling a word token from the topic-specific word distribution for the chosen topic $\textsc{Multi}(\phi_z)$. Latent Dirichlet Allocation (LDA) (Blei et al., 2003) generalizes PLSA to a proper generative model for documents and places Dirichlet priors over the parameters θ and ϕ. In the experiments in this paper, our implementation of PLSA is LDA with maximum a posteriori (MAP) inference, which was shown to be comparable to the current best Bayesian inference methods for LDA (Asuncion et al., 2009).

Recently, these topic models have been generalized to handle pairs or tuples of corresponding documents, which could be translations in multiple languages, or documents in the same language that are considered similar. For instance, the Poly-lingual Topic Model (PLTM) (Mimno et al., 2009) is an extension to LDA that views documents in a tuple as having a shared topic vector θ. Each of the documents in the tuple uses θ to select the topics z of tokens, but could use a different (language-specific) word-topic-distribution $\textsc{Multi}(\phi_z^L)$. Two additional models, Joint PLSA (JPLSA) and Coupled PLSA (CPLSA) were introduced in (Platt et al., 2010). JPLSA is a close variant of PLTM when documents of all languages share the same word-topic distribution parameters, and MAP inference is performed instead of Bayesian. CPLSA extends JPLSA by constraining paired documents to not only share the same prior topic distribution θ, but to also have similar fractions of tokens assigned to each topic. This constraint is enforced on expectation using posterior regularization (Ganchev et al., 2009).

2.2 Linear Projection Methods

The earliest method for projecting term vectors into a low-dimensional concept space is Latent Semantic Analysis (LSA) (Deerwester et al., 1990). LSA models all documents in a corpus using a $n \times d$ document-term matrix $\mathbf{D}$ and performs singular

value decomposition (SVD) on $\mathbf{D}$. The k biggest singular values are then used to find the $d \times k$ projection matrix. Instead of SVD, LSA can be done by applying eigen-decomposition on the correlation matrix between terms $\mathbf{C} = \mathbf{D}^T\mathbf{D}$. This is very similar to principal component analysis (PCA), where a covariance matrix between terms is used. In practice, term vectors are very sparse and their means are close to 0. Therefore, the correlation matrix is in fact close to the covariance matrix.

To model pairs of comparable documents, LSA/PCA has been extended in different ways. For instance, Cross-language Latent Semantic Indexing (CL-LSI) (Dumais et al., 1997) applies LSA to concatenated comparable documents from different languages. Oriented Principal Component Analysis (OPCA) (Diamantaras and Kung, 1996; Platt et al., 2010) solves a generalized eigen problem by introducing a noise covariance matrix to ensure that comparable documents can be projected closely. Canonical Correlation Analysis (CCA) (Vinokourov et al., 2003) finds projections that maximize the cross-covariance between the projected vectors.

2.3 Distance Metric Learning

Measuring the similarity between two vectors can be viewed as equivalent to measuring their distance, as the cosine score has a bijection mapping to the Euclidean distance of unit vectors. Most work on metric learning learns a Mahalanobis distance, which generalizes the standard squared Euclidean distance by modeling the similarity of elements in different dimensions using a positive semi-definite matrix $\mathbf{A}$. Given two vectors $\mathbf{x}$ and $\mathbf{y}$, their squared Mahalanobis distance is: $d_{\mathbf{A}} = (\mathbf{x} - \mathbf{y})^T\mathbf{A}(\mathbf{x} - \mathbf{y})$. However, the computational complexity of learning a general Mahalanobis matrix is at least $O(n^2)$, where n is the dimensionality of the input vectors. Therefore, such methods are not practical for high dimensional problems in the text domain.

In order to tackle this issue, special metric learning approaches for high-dimensional spaces have been proposed. For example, high dimension low-rank (HDLR) metric learning (Davis and Dhillon, 2008) constrains the form of $\mathbf{A} = \mathbf{U}\mathbf{U}^T$, where $\mathbf{U}$ is similar to the regular projection matrix, and adapts information-theoretic metric learning (ITML) (Davis et al., 2007) to learn $\mathbf{U}$.

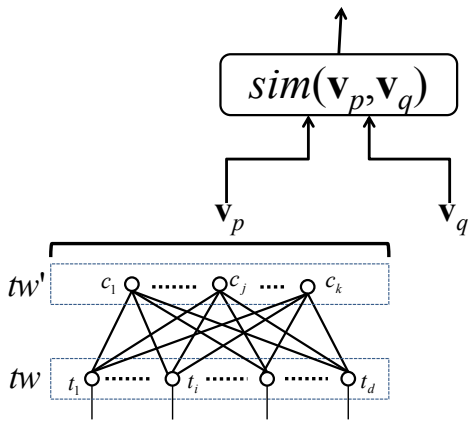

Figure 1: Learning concept vectors. The output layer consists of a small number of concept nodes, where the weight of each node is a linear combination of all the original term weights.

3 Similarity Learning via Siamese Neural Network (S2Net)

Given pairs of documents with their labels, such as binary or real-valued similarity scores, our goal is to construct a projection matrix that maps the corresponding term-vectors into a low-dimensional *concept* space such that similar documents are close when projected into this space. We propose a similarity learning framework via Siamese neural network (S2Net) to learn the projection matrix *directly* from labeled data. In this section, we introduce its model design and describe the training process.

3.1 Model Design

The network structure of S2Net consists of two layers. The input layer corresponds to the raw term vector, where each node represents a term in the original vocabulary and its associated value is determined by a term-weighting function such as TFIDF. The output layer is the learned low-dimensional vector representation that captures relationships among terms. Similarly, each node of the output layer is an element in the new concept vector. In this work, the final similarity score is calculated using the cosine function, which is the standard choice for document similarity (Manning et al., 2008). Our framework can be easily extended to other similarity functions as long as they are differentiable.

The output of each concept node is a linear com-

bination of the weights of all the terms in the original term vector. In other words, these two layers of nodes form a complete bipartite graph as shown in Fig. 1. The output of a concept node c_j is thus defined as:

$$tw'(c_j) = \sum_{t_i \in V} \alpha_{ij} \cdot tw(t_i) \quad (1)$$

Notice that it is straightforward to add a non-linear activation function (e.g., sigmoid) in Eq. (1), which can potentially lead to better results. However, in the current design, the model form is exactly the same as the low-rank projection matrix derived by PCA, OPCA or CCA, which facilitates comparison to alternative projection methods. Using concise matrix notation, let $\mathbf{f}$ be a raw d-by-1 term vector, $\mathbf{A} = [\alpha_{ij}]_{d \times k}$ the projection matrix. $\mathbf{g} = \mathbf{A}^T\mathbf{f}$ is thus the k-by-1 projected concept vector.

3.2 Loss Function and Training Procedure

For a pair of term vectors $\mathbf{f}_p$ and $\mathbf{f}_q$, their similarity score is defined by the cosine value of the corresponding concept vectors $\mathbf{g}_p$ and $\mathbf{g}_q$ according to the projection matrix $\mathbf{A}$.

$$sim_{\mathbf{A}}(\mathbf{f}_p, \mathbf{f}_q) = \frac{\mathbf{g}_p^T \mathbf{g}_q}{||\mathbf{g}_p|| ||\mathbf{g}_q||},$$

where $\mathbf{g}_p = \mathbf{A}^T\mathbf{f}_p$ and $\mathbf{g}_q = \mathbf{A}^T\mathbf{f}_q$. Let y_{pq} be the true label of this pair. The loss function can be as simple as the mean-squared error $\frac{1}{2}(y_{pq} - sim_{\mathbf{A}}(\mathbf{f}_p, \mathbf{f}_q))^2$. However, in many applications, the similarity scores are used to select the closest text objects given the query. For example, given a query document, we only need to have the comparable document in the target language ranked higher than any other documents. In this scenario, it is more important for the similarity measure to yield a good ordering than to match the target similarity scores. Therefore, we use a pairwise learning setting by considering a pair of similarity scores (i.e., from two vector pairs) in our learning objective.

Consider two pairs of term vectors $(\mathbf{f}_{p_1}, \mathbf{f}_{q_1})$ and $(\mathbf{f}_{p_2}, \mathbf{f}_{q_2})$, where the first pair has higher similarity. Let Δ be the difference of their similarity scores. Namely, $\Delta = sim_{\mathbf{A}}(\mathbf{f}_{p_1}, \mathbf{f}_{q_1}) - sim_{\mathbf{A}}(\mathbf{f}_{p_2}, \mathbf{f}_{q_2})$. We use the following logistic loss over Δ, which upper-bounds the pairwise accuracy (i.e., 0-1 loss):

$$L(\Delta; \mathbf{A}) = \log(1 + exp(-\gamma\Delta)) \quad (2)$$

Because of the cosine function, we add a scaling factor γ that magnifies Δ from $[-2, 2]$ to a larger range, which helps penalize more on the prediction errors. Empirically, the value of γ makes no difference as long as it is large enough[1]. In the experiments, we set the value of γ to 10. Optimizing the model parameters $\mathbf{A}$ can be done using gradient based methods. We derive the gradient of the whole batch and apply the quasi-Newton optimization method L-BFGS (Nocedal and Wright, 2006) directly. For a cleaner presentation, we detail the gradient derivation in Appendix A. Given that the optimization problem is not convex, initializing the model from a good projection matrix often helps reduce training time and may lead to convergence to a better local minimum. Regularization can be done by adding a term $\frac{\beta}{2}||\mathbf{A} - \mathbf{A}_0||^2$ in Eq. (2), which forces the learned model not to deviate too much from the starting point ($\mathbf{A}_0$), or simply by early stopping. Empirically we found that the latter is more effective and it is used in the experiments.

4 Experiments

We compare S2Net experimentally with existing approaches on two very different tasks: cross-lingual document retrieval and ad relevance measures.

4.1 Comparable Document Retrieval

With the growth of multiple languages on the Web, there is an increasing demand of processing cross-lingual documents. For instance, machine translation (MT) systems can benefit from training on sentences extracted from parallel or comparable documents retrieved from the Web (Munteanu and Marcu, 2005). Word-level translation lexicons can also be learned from comparable documents (Fung and Yee, 1998; Rapp, 1999). In this cross-lingual document retrieval task, given a query document in one language, the goal is to find the most *similar* document from the corpus in another language.

4.1.1 Data & Setting

We followed the comparable document retrieval setting described in (Platt et al., 2010) and evaluated S2Net on the Wikipedia dataset used in that paper. This data set consists of Wikipedia documents

[1] Without the γ parameter, the model still outperforms other baselines in our experiments, but with a much smaller gain.

in two languages, English and Spanish. An article in English is paired with a Spanish article if they are identified as comparable across languages by the Wikipedia community. To conduct a fair comparison, we use the same term vectors and data split as in the previous study. The numbers of document pairs in the training/development/testing sets are 43,380, 8,675 and 8,675, respectively. The dimensionality of the raw term vectors is 20,000.

The models are evaluated by using each English document as query against all documents in Spanish and *vice versa*; the results from the two directions are averaged. Performance is evaluated by two metrics: the Top-1 accuracy, which tests whether the document with the highest similarity score is the true comparable document, and the Mean Reciprocal Rank (MRR) of the true comparable.

When training the S2Net model, all the comparable document pairs are treated as positive examples and all other pairs are used as negative examples. Naively treating these 1.8 billion pairs (i.e., 43380^2) as independent examples would make the training very inefficient. Fortunately, most computation in deriving the batch gradient can be reused via compact matrix operations and training can still be done efficiently. We initialized the S2Net model using the matrix learned by OPCA, which gave us the best performance on the development set[2].

Our approach is compared with most methods studied in (Platt et al., 2010), including the best performing one. For CL-LSI, OPCA, and CCA, we include results from that work directly. In addition, we re-implemented and improved JPLSA and CPLSA by changing three settings: we used separate vocabularies for the two languages as in the Poly-lingual topic model (Mimno et al., 2009), we performed 10 EM iterations for folding-in instead of only one, and we used the Jensen-Shannon distance instead of the L1 distance. We also attempted to apply the HDLR algorithm. Because this algorithm does not scale well as the number of training examples increases, we used 2,500 positive and 2,500 negative document pairs for training. Unfortunately, among all the hyper-parameter settings we tested, HDLR could not outperform its initial model, which was the OPCA matrix. Therefore we omit these results.

[2]S2Net outperforms OPCA when initialized from a random or CL-LSI matrix, but with a smaller gain. For example, when the number of dimensions is 1000, the MRR score of OPCA is 0.7660. Starting from the CL-LSI and OPCA matrices, the MRR scores of S2Net are 0.7745 and 0.7855, respectively.

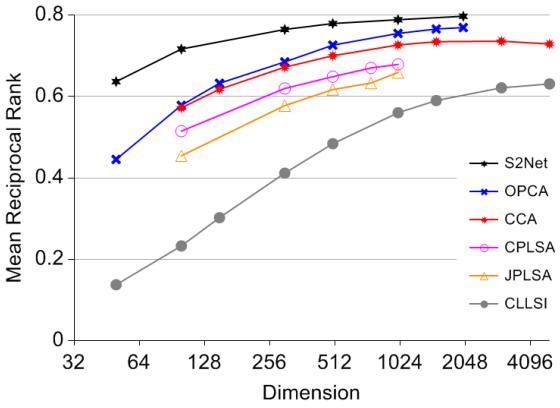

Figure 2: Mean reciprocal rank versus dimension for Wikipedia. Results of OPCA, CCA and CL-LSI are from (Platt et al., 2010).

4.1.2 Results

Fig. 2 shows the MRR performance of all methods on the development set, across different dimensionality settings of the concept space. As can be observed from the figure, higher dimensions usually lead to better results. In addition, S2Net consistently performs better than all other methods across different dimensions. The gap is especially large when projecting input vectors to a low-dimensional space, which is preferable for efficiency. For instance, using 500 dimensions, S2Net already performs as well as OPCA with 2000 dimensions.

Table 1 shows the averaged Top-1 accuracy and MRR scores of all methods on the test set, where the dimensionality for each method is optimized on the development set (Fig. 2). S2Net clearly outperforms all other methods and the difference in terms of accuracy is statistically significant[3].

4.2 Ad Relevance

Paid search advertising is the main revenue source that supports modern commercial search engines. To ensure satisfactory user experience, it is important to provide both relevant ads and regular search

[3]We use the unpaired t-test with Bonferroni correction and the difference is considered statistically significant when the p-value is less than 0.01.

Algorithm	Dimension	Accuracy	MRR
S2Net	2000	**0.7447**	**0.7973**
OPCA	2000	0.7255	0.7734
CCA	1500	0.6894	0.7378
CPLSA	1000	0.6329	0.6842
JPLSA	1000	0.6079	0.6604
CL-LSI	5000	0.5302	0.6130

Table 1: Test results for comparable document retrieval in Wikipedia. Results of OPCA, CCA and CL-LSI are from (Platt et al., 2010).

results. Previous work on ad relevance focuses on constructing appropriate term-vectors to represent queries and ad-text (Broder et al., 2008; Choi et al., 2010). In this section, we extend the work in (Yih and Jiang, 2010) and show how S2Net can exploit annotated query–ad pairs to improve the vector representation in this monolingual setting.

4.2.1 Data & Tasks

The ad relevance dataset we used consists of 12,481 unique queries randomly sampled from the logs of the Bing search engine. For each query, a number of top ranked ads are selected, which results in a total number of 567,744 query-ad pairs in the dataset. Each query-ad pair is manually labeled as *same*, *subset*, *superset* or *disjoint*. In our experiment, when the task is a binary classification problem, pairs labeled as *same*, *subset*, or *superset* are considered relevant, and pairs labeled as *disjoint* are considered irrelevant. When pairwise comparisons are needed in either training or evaluation, the relevance order is *same* $>$ *subset* $=$ *superset* $>$ *disjoint*. The dataset is split into training (40%), validation (30%) and test (30%) sets by queries.

Because a query string usually contains only a few words and thus provides very little content, we applied the same web relevance feedback technique used in (Broder et al., 2008) to create "pseudo-documents" to represent queries. Each query in our data set was first issued to the search engine. The result page with up to 100 snippets was used as the pseudo-document to create the raw term vectors. On the ad side, we used the ad landing pages instead of the short ad-text. Our vocabulary set contains 29,854 words and is determined using a document frequency table derived from a large collection of Web documents. Only words with counts larger than a pre-selected threshold are retained.

How the data is used in training depends on the model. For S2Net, we constructed preference pairs in the following way. For the same query, each relevant ad is paired with a less relevant ad. The loss function from Eq. (2) encourages achieving a higher similarity score for the more relevant ad. For HDLR, we used a sample of 5,000 training pairs of queries and ads, as it was not able to scale to more training examples. For OPCA, CCA, PLSA and JPLSA, we constructed a parallel corpus using only relevant pairs of queries and ads, as the negative examples (irrelevant pairs of queries and ads) cannot be used by these models. Finally, PCA and PLSA learn the models from all training queries and documents without using any relevance information.

We tested S2Net and other methods in two different application scenarios. The first is to use the ad relevance measure as an *ad filter*. When the similarity score between a query and an ad is below a preselected decision threshold, this ad is considered irrelevant to the query and will be filtered. Evaluation metrics used for this scenario are the ROC analysis and the area under the curve (AUC). The second one is the ranking scenario, where the ads are selected and ranked by their relevance scores. In this scenario, the performance is evaluated by the standard ranking metric, *Normalized Discounted Cumulative Gain* (NDCG) (Jarvelin and Kekalainen, 2000).

4.2.2 Results

We first compare different methods in their AUC and NDCG scores. *TFIDF* is the basic term vector representation with the TFIDF weighting ($tf \cdot \log(N/df)$). It is used as our baseline and also as the raw input for S2Net, HDLR and other linear projection methods. Based on the results on the development set, we found that PCA performs better than OPCA and CCA. Therefore, we initialized the models of S2Net and HDLR using the PCA matrix. Table 2 summarizes results on the test set. All models, except TFIDF, use 1000 dimensions and their best configuration settings selected on the validation set.

TFIDF is a very strong baseline on this monolingual ad relevance dataset. Among all the methods we tested, at dimension 1000, only S2Net outperforms the raw TFIDF cosine measure in every evaluation metric, and the difference is statistically sig-

	AUC	NDCG@1	NDCG@3	NDCG@5
S2Net	**0.892**	**0.855**	**0.883**	**0.901**
TFIDF	0.861	0.825	0.854	0.876
HDLR	0.855	0.826	0.856	0.877
CPLSA	0.853	0.845	0.872	0.890
PCA	0.848	0.815	0.847	0.870
OPCA	0.844	0.817	0.850	0.872
JPLSA	0.840	0.838	0.864	0.883
CCA	0.836	0.820	0.852	0.874
PLSA	0.835	0.831	0.860	0.879

Table 2: The AUC and NDCG scores of the cosine similarity scores on different vector representations. The dimension for all models except TFIDF is 1000.

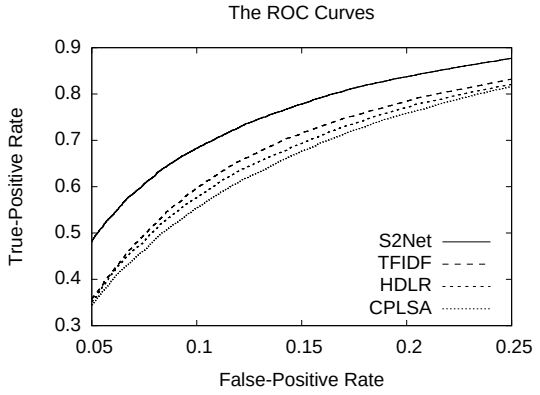

Figure 3: The ROC curves of S2Net, TFIDF, HDLR and CPLSA when the similarity scores are used as ad filters.

nificant[4]. In contrast, both CPLSA and HDLR have higher NDCG scores but lower AUC values, and OPCA/CCA perform roughly the same as PCA.

When the cosine scores of these vector representations are used as ad filters, their ROC curves (focusing on the low false-positive region) are shown in Fig. 3. It can be clearly observed that the similarity score computed based on vectors derived from S2Net indeed has better quality, compared to the raw TFIDF representation. Unfortunately, other approaches perform worse than TFIDF and their performance in the low false-positive region is consistent with the AUC scores.

Although ideally we would like the dimensionality of the projected concept vectors to be as small as possible for efficient processing, the quality of the concept vector representation usually degrades as well. It is thus interesting to know the best trade-off point between these two variables. Table 3 shows the AUC and NDCG scores of S2Net at different dimensions, as well as the results achieved by TFIDF and PCA, HDLR and CPLSA at 1000 dimensions. As can be seen, S2Net surpasses TFIDF in AUC at dimension 300 and keeps improving as the dimensionality increases. Its NDCG scores are also consistently higher across all dimensions.

4.3 Discussion

It is encouraging to find that S2Net achieves strong performance in two very different tasks, given that it is a conceptually simple model. Its empirical success can be attributed to two factors. First, it is flexible in choosing the loss function and constructing training examples and is thus able to optimize the model directly for the target task. Second, it can be trained on a large number of examples. For example, HDLR can only use a few thousand examples and is not able to learn a matrix better than its initial model for the task of cross-lingual document retrieval. The fact that linear projection methods like OPCA/CCA and generative topic models like JPLSA/CPLSA cannot use negative examples more effectively also limits their potential.

In terms of scalability, we found that methods based on eigen decomposition, such as PCA, OPCA and CCA, take the least training time. The complexity is decided by the size of the covariance matrix, which is quadratic in the number of dimensions. On a regular eight-core server, it takes roughly 2 to 3 hours to train the projection matrix in both experiments. The training time of S2Net scales roughly linearly to the number of dimensions and training examples. In each iteration, performing the projection takes the most time in gradient derivation, and the complexity is $O(mnk)$, where m is the number of distinct term-vectors, n is the largest number of non-zero elements in the sparse term-vectors and k is the dimensionality of the concept space. For cross-lingual document retrieval, when $k = 1000$, each iteration takes roughly 48 minutes and about 80 iterations are required to convergence. Fortunately, the gradient computation is easily parallelizable and further speed-up can be achieved using a cluster.

[4]For AUC, we randomly split the data into 50 subsets and ran a paired-t test between the corresponding AUC scores. For NDCG, we compared the DCG scores per query of the compared models using the paired-t test. The difference is considered statistically significant when the p-value is less than 0.01.

	TFIDF	HDLR	CPLSA	PCA	$S2Net_{100}$	$S2Net_{300}$	$S2Net_{500}$	$S2Net_{750}$	$S2Net_{1000}$
AUC	0.861	0.855	0.853	0.848	0.855	0.879	0.880	0.888	0.892
NDCG@1	0.825	0.826	0.845	0.815	0.843	0.852	0.856	0.860	0.855
NDCG@3	0.854	0.856	0.872	0.847	0.871	0.879	0.881	0.884	0.883
NDCG@5	0.876	0.877	0.890	0.870	0.890	0.897	0.899	0.902	0.901

Table 3: The AUC and NDCG scores of S2Net at different dimensions. PCA, HDLR & CPLSA (at dimension 1000) along with the raw TFIDF representation are used for reference.

5 Related Work

Although the high-level design of S2Net follows the Siamese architecture (Bromley et al., 1993; Chopra et al., 2005), the network construction, loss function and training process of S2Net are all different compared to previous work. For example, targeting the application of face verification, Chopra et al. (2005) used a convolutional network and designed a contrastive loss function for optimizing a Euclidean distance metric. In contrast, the network of S2Net is equivalent to a linear projection matrix and has a pairwise loss function. In terms of the learning framework, S2Net is closely related to several neural network based approaches, including autoencoders (Hinton and Salakhutdinov, 2006) and finding low-dimensional word representations (Collobert and Weston, 2008; Turian et al., 2010). Architecturally, S2Net is also similar to RankNet (Burges et al., 2005), which can be viewed as a Siamese neural network that learns a ranking function.

The strategy that S2Net takes to learn from labeled pairs of documents can be analogous to the work of distance metric learning. Although high dimensionality is not a problem to algorithms like HDLR, it suffers from a different scalability issue. As we have observed in our experiments, the algorithm can only handle a small number of similarity/dissimilarity constraints (i.e., the labeled examples), and is not able to use a large number of examples to learn a better model. Empirically, we also found that HDLR is very sensitive to the hyperparameter settings and its performance can vary substantially from iteration to iteration.

Other than the applications presented in this paper, concept vectors have shown useful in traditional IR tasks. For instance, Egozi et al. (2008) use *explicit semantic analysis* to improve the retrieval recall by leveraging Wikipedia. In a companion paper, we also demonstrated that various topic models including S2Net can enhance the ranking function (Gao et al., 2011). For text categorization, similarity between terms is often encoded as kernel functions embedded in the learning algorithms, and thus increase the classification accuracy. Representative approaches include *latent semantic kernels* (Cristianini et al., 2002), which learns an LSA-based kernel function from a document collection, and work that computes term-similarity based on the linguistic knowledge provided by WordNet (Basili et al., 2005; Bloehdorn and Moschitti, 2007).

6 Conclusions

In this paper, we presented S2Net, a discriminative approach for learning a projection matrix that maps raw term-vectors to a low-dimensional space. Our learning method directly optimizes the model so that the cosine score of the projected vectors can become a reliable similarity measure. The strength of this model design has been shown empirically in two very different tasks. For cross-lingual document retrieval, S2Net significantly outperforms OPCA, which is the best prior approach. For ad selection and filtering, S2Net also outperforms all methods we compared it with and is the only technique that beats the raw TFIDF vectors in both AUC and NDCG.

The success of S2Net is truly encouraging, and we would like to explore different directions to further enhance the model in the future. For instance, it will be interesting to extend the model to learn non-linear transformations. In addition, since the pairs of text objects being compared often come from different distributions (e.g., English documents vs. Spanish documents or queries vs. pages), learning two different matrices instead of one could increase the model expressivity. Finally, we would like to apply S2Net to more text similarity tasks, such as word similarity and entity recognition and discovery.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of HLT-NAACL*, pages 19–27, June.

Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. 2009. On smoothing and inference for topic models. In *UAI*.

Roberto Basili, Marco Cammisa, and Alessandro Moschitti. 2005. Effective use of WordNet semantics via kernel-based learning. In *CoNLL*.

David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Stephan Bloehdorn and Alessandro Moschitti. 2007. Combined syntactic and semantic kernels for text classification. In *ECIR*, pages 307–318.

Andrei Z. Broder, Peter Ciccolo, Marcus Fontoura, Evgeniy Gabrilovich, Vanja Josifovski, and Lance Riedel. 2008. Search advertising using web relevance feedback. In *CIKM*, pages 1013–1022.

Jane Bromley, James W. Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "Siamese" time delay neural network. *International Journal Pattern Recognition and Artificial Intelligence*, 7(4):669–688.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *ICML*.

Y. Choi, M. Fontoura, E. Gabrilovich, V. Josifovski, M. Mediano, and B. Pang. 2010. Using landing pages for sponsored search ad selection. In *WWW*.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of CVPR-2005*, pages 539–546.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML*.

Nello Cristianini, John Shawe-Taylor, and Huma Lodhi. 2002. Latent semantic kernels. *Journal of Intelligent Information Systems*, 18(2–3):127–152.

Jason V. Davis and Inderjit S. Dhillon. 2008. Structured metric learning for high dimensional problems. In *KDD*, pages 195–203.

Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. 2007. Information-theoretic metric learning. In *ICML*.

Scott Deerwester, Susan Dumais, George Furnas, Thomas Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Konstantinos I. Diamantaras and S.Y. Kung. 1996. *Principal Component Neural Networks: Theory and Applications*. Wiley-Interscience.

Susan T. Dumais, Todd A. Letsche, Michael L. Littman, and Thomas K. Landauer. 1997. Automatic cross-linguistic information retrieval using latent semantic indexing. In *AAAI-97 Spring Symposium Series: Cross-Language Text and Speech Retrieval*.

Ofer Egozi, Evgeniy Gabrilovich, and Shaul Markovitch. 2008. Concept-based feature generation and selection for information retrieval. In *AAAI*.

Pascale Fung and Lo Yuen Yee. 1998. An IR approach for translating new words from nonparallel, comparable texts. In *Proceedings of COLING-ACL*.

Kuzman Ganchev, Joao Graca, Jennifer Gillenwater, and Ben Taskar. 2009. Posterior regularization for structured latent variable models. Technical Report MS-CIS-09-16, University of Pennsylvania.

Jianfeng Gao, Kristina Toutanova, and Wen-tau Yih. 2011. Clickthrough-based latent semantic models for web search. In *SIGIR*.

G. E. Hinton and R. R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *SIGIR '99*, pages 50–57.

K. Jarvelin and J. Kekalainen. 2000. Ir evaluation methods for retrieving highly relevant documents. In *SIGIR*, pages 41–48.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proc. of COLING-ACL 98*.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Pres.

David Mimno, Hanna W. Wallach, Jason Naradowsky, David A. Smith, and Andrew McCallum. 2009. Polylingual topic models. In *EMNLP*.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31:477–504.

Jorge Nocedal and Stephen Wright. 2006. *Numerical Optimization*. Springer, 2nd edition.

John Platt, Kristina Toutanova, and Wen-tau Yih. 2010. Translingual document representations from discriminative projections. In *EMNLP*.

Reinhard Rapp. 1999. Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of the ACL*, pages 519–526.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *ACL*.

Alexei Vinokourov, John Shawe-taylor, and Nello Cristianini. 2003. Inferring a semantic representation of text via cross-language correlation analysis. In *NIPS-15*.

Vishnu Vyas and Patrick Pantel. 2009. Semi-automatic entity set refinement. In *NAACL '09*, pages 290–298.

Wen-tau Yih and Ning Jiang. 2010. Similarity models for ad relevance measures. In *MLOAD - NIPS 2010 Workshop on online advertising*.

Appendix A. Gradient Derivation

The gradient of the loss function in Eq. (2) can be derived as follows.

$$\frac{\partial L(\Delta, \mathbf{A})}{\partial \mathbf{A}} = \frac{-\gamma}{1+\exp(-\gamma\Delta)}\frac{\partial \Delta}{\partial \mathbf{A}}$$
$$\frac{\partial \Delta}{\partial \mathbf{A}} = \frac{\partial}{\partial \mathbf{A}} sim_{\mathbf{A}}(\mathbf{f}_{p_1}, \mathbf{f}_{q_1}) - \frac{\partial}{\partial \mathbf{A}} sim_{\mathbf{A}}(\mathbf{f}_{p_2}, \mathbf{f}_{q_2})$$
$$\frac{\partial}{\partial \mathbf{A}} sim_{\mathbf{A}}(\mathbf{f}_p, \mathbf{f}_q) = \frac{\partial}{\partial \mathbf{A}} \cos(\mathbf{g}_p, \mathbf{g}_q),$$

where $\mathbf{g}_p = \mathbf{A}^T\mathbf{f}_p$ and $\mathbf{g}_q = \mathbf{A}^T\mathbf{f}_q$ are the projected concept vectors of $\mathbf{f}_q$ and $\mathbf{f}_q$. The gradient of the cosine score can be further derived in the following steps.

$$\begin{aligned}
\cos(\mathbf{g}_p, \mathbf{g}_q) &= \frac{\mathbf{g}_p^T\mathbf{g}_q}{\|\mathbf{g}_p\|\|\mathbf{g}_q\|} \\
\nabla_{\mathbf{A}} \mathbf{g}_p^T\mathbf{g}_q &= (\nabla_{\mathbf{A}}\mathbf{A}^T\mathbf{f}_p)\mathbf{g}_q + (\nabla_{\mathbf{A}}\mathbf{A}^T\mathbf{f}_q)\mathbf{g}_p \\
&= \mathbf{f}_p\mathbf{g}_q^T + \mathbf{f}_q\mathbf{g}_p^T \\
\nabla_{\mathbf{A}} \frac{1}{\|\mathbf{g}_p\|} &= \nabla_{\mathbf{A}}(\mathbf{g}_p^T\mathbf{g}_p)^{-\frac{1}{2}} \\
&= -\frac{1}{2}(\mathbf{g}_p^T\mathbf{g}_p)^{-\frac{3}{2}}\nabla_{\mathbf{A}}(\mathbf{g}_p^T\mathbf{g}_p) \\
&= -(\mathbf{g}_p^T\mathbf{g}_p)^{-\frac{3}{2}}\mathbf{f}_p\mathbf{g}_p^T \\
\nabla_{\mathbf{A}} \frac{1}{\|\mathbf{g}_q\|} &= -(\mathbf{g}_q^T\mathbf{g}_q)^{-\frac{3}{2}}\mathbf{f}_q\mathbf{g}_q^T
\end{aligned}$$

Let a, b, c be $\mathbf{g}_p^T\mathbf{g}_q$, $1/\|\mathbf{g}_p\|$ and $1/\|\mathbf{g}_q\|$, respectively.

$$\begin{aligned}
\nabla_{\mathbf{A}} \frac{\mathbf{g}_p^T\mathbf{g}_q}{\|\mathbf{g}_p\|\|\mathbf{g}_q\|} = & -abc^3\mathbf{f}_q\mathbf{g}_q^T - acb^3\mathbf{f}_p\mathbf{g}_p^T \\
& + bc(\mathbf{f}_p\mathbf{g}_q^T + \mathbf{f}_q\mathbf{g}_p^T)
\end{aligned}$$