Weiwei: A Simple Unsupervised Latent Semantics based Approach for Sentence Similarity

Weiwei Guo
Department of Computer Science,
Columbia University,
weiwei@cs.columbia.edu

Mona Diab
Center for Computational Learning Systems,
Columbia University,
mdiab@ccls.columbia.edu

Abstract

The Semantic Textual Similarity (STS) shared task (Agirre et al., 2012) computes the degree of semantic equivalence between two sentences.[1] We show that a simple unsupervised latent semantics based approach, Weighted Textual Matrix Factorization that only exploits bag-of-words features, can outperform most systems for this task. The key to the approach is to carefully handle missing words that are not in the sentence, and thus rendering it superior to Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA). Our system ranks 20 out of 89 systems according to the official evaluation metric for the task, Pearson correlation, and it ranks 10/89 and 19/89 in the other two evaluation metrics employed by the organizers.

1 Introduction

Identifying the degree of semantic similarity [SS] between two sentences is helpful for many NLP topics. In Machine Translation (Kauchak and Barzilay, 2006) and Text Summarization (Zhou et al., 2006), results are automatically evaluated based on sentence comparison. In Text Coherence Detection (Lapata and Barzilay, 2005), sentences are linked together by similar or related words. For Word Sense Disambiguation, researchers (Banerjee and Pedersen, 2003; Guo and Diab, 2012a) construct a sense similarity measure from the sentence similarity of the sense definitions.

Almost all SS approaches decompose the task into word pairwise similarity problems. For example, Islam and Inkpen (2008) create a matrix for each sentence pair, where columns are the words in the first sentence and rows are the words in the second sentence, and each cell stores the distributional similarity of the two words. Then they create an alignment between words in two sentences, and sentence similarity is calculated based on the sum of the similarity of aligned word pairs. There are two disadvantages with word similarity based approaches: 1. lexical ambiguity as the word pairwise similarity ignores the semantic interaction between the word and sentence/context. 2. word co-occurrence information is not as sufficiently exploited as they are in latent variable models such as Latent Semantic Analysis (LSA) (Landauer et al., 1998) and Latent Dirichilet Allocation (LDA) (Blei et al., 2003). On the other hand, latent variable models can solve the two issues naturally by modeling the semantics of words and sentences simultaneously in the low-dimensional latent space.

However, attempts at addressing SS using LSA perform significantly below word similarity based models (Mihalcea et al., 2006; O'Shea et al., 2008). We believe the reason is that the observed words in a sentence are too few for latent variable models to learn robust semantics. For example, given the two sentences of WordNet sense definitions for $bank\#n\#1$ and $stock\#n\#1$:

bank#n#1: *a financial institution that accepts deposits and channels the money into lending activities*

stock#n#1: *the capital raised by a corporation through the issue of shares entitling holders to an ownership interest (equity)*

LDA can only find the dominant topic (the $financial$ topic) based on the observed words without further discernibility. In this case, many sen-

[1] Mona Diab, co-author of this paper, is one of the task organizers

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 586–590,
Montréal, Canada, June 7-8, 2012.

tences will share the same latent semantics profile, as long as they are in the same topic/domain.

In our work (Guo and Diab, 2012b), we propose to model the missing words (words that are not in the sentence) to address the sparseness issue for the SS task. Our intuition is since observed words in a sentence are too few to tell us what the sentence is about, missing words can be used to tell us what the sentence is **not** about. We assume that the semantic space of both the observed and missing words make up the complete semantic profile of a sentence. We implement our idea using a weighted matrix factorization approach (Srebro and Jaakkola, 2003), which allows us to treat observed words and missing words differently.

It should be noted that our approach is very general (similar to LSA/LDA) in that it can be applied to any genre of short texts, in a manner different from existing work that models short texts by using additional data, e.g., Ramage et al. (2010) model tweets using their metadata (author, hashtag, etc). Also we do not extract additional features such as multiwords expression or syntax from sentences – all we use is bag-of-words feature.

2 Related Work

Almost all current SS methods work in the high-dimensional word space, and rely heavily on word/sense similarity measures. The word/sense similarity measure is either knowledge based (Li et al., 2006; Feng et al., 2008; Ho et al., 2010; Tsatsaronis et al., 2010), corpus-based (Islam and Inkpen, 2008) or hybrid (Mihalcea et al., 2006). Almost all of them are evaluated on a data set introduced in (Li et al., 2006). The LI06 data set consists of 65 pairs of noun definitions selected from the Collin Cobuild Dictionary. A subset of 30 pairs is further selected by LI06 to render the similarity scores evenly distributed. Our approach has outperformed most of the previous methods on LI06 achieving the second best Pearson's correlation and the best Spearman correlation (Guo and Diab, 2012b).

3 Learning Latent Semantics of Sentences

3.1 Intuition

Given only a few observed words in a sentence, there are many hypotheses of latent vectors that are highly related to the observed words. Therefore, missing words can be used to prune the hypotheses that are also highly related to the missing words.

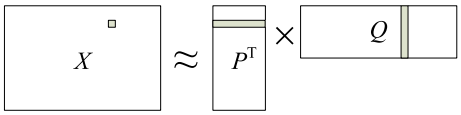

Figure 1: Matrix Factorization

Consider the hypotheses of latent vectors in Table 1 for the sentence of the WordNet definition of $bank\#n\#1$. Assume there are 3 dimensions in our latent model: *financial, sport, institution*. We use R_o^v to denote the sum of relatedness between latent vector v and all observed words; similarly, R_m^v is the sum of relatedness between the vector v and all missing words. Hypothesis v_1 is given by topic models, where only the $financial$ sentence is found, and it has the maximum relatedness to observed words in $bank\#n\#1$ sentence $R_o^{v_1}$=20. v_2 is the ideal latent vector, since it also detects that $bank\#n\#1$ is related to $institution$. It has a slightly smaller $R_o^{v_2}$=18, but more importantly, relatedness to missing words $R_m^{v_2}$=300 is substantially smaller than $R_m^{v_1}$=600.

However, we cannot simply choose a hypothesis with the maximum $R_o - R_m$ value, since v_3, which is clearly not related to $bank\#n\#1$ but with a minimum R_m=100, will be our final answer. The solution is straightforward: give a smaller weight to missing words, e.g., so that the algorithm tries to select a hypothesis with maximum value of $R_o - 0.01 \times R_m$. To implement this idea, we model the missing words in the weighted matrix factorization framework [WMF] (Srebro and Jaakkola, 2003).

3.2 Modeling Missing Words by Weighted Matrix Factorization

Given a corpus we represent the corpus as an $M \times N$ matrix X. The row entries of the matrix are the unique N words in the corpus, and the M columns are the sentence ids of all the sentences. The yielded $N \times M$ co-occurrence matrix X comprises the TF-IDF values in each X_{ij} cell, namely that TF-IDF value of word w_i in sentence s_j.

In WMF, the original matrix X is factorized into two matrices such that $X \approx P^\top Q$, where P is a $K \times M$ matrix, and Q is a $K \times N$ matrix (Figure 1). In this scenario, the latent semantics of each word w_i or sentence s_j is represented as a K-dimension vector

	financial	**sport**	**institution**	R_o	R_m	$R_o - R_m$	$R_o - 0.01R_m$
v_1	1	0	0	***20***	***600***	-580	14
v_2	0.6	0	0.1	18	300	-282	***15***
v_3	0.2	0.3	0.2	5	100	***-95***	4

Table 1: Three possible hypotheses of latent vectors for definition of $bank\#n\#1$

$P_{\cdot,i}$ or $Q_{\cdot,j}$. Note that the inner product of $P_{\cdot,i}$ and $Q_{\cdot,j}$ is used to approximate the semantic relatedness of word w_i and sentence s_j: $X_{ij} \approx P_{\cdot,i} \cdot Q_{\cdot,j}$, as the shaded parts in Figure 1.

In WMF each cell is associated with a weight, so missing words cells (X_{ij}=0) can have a much less contribution than observed words. Assume w_m is the weight for missing words cells. The latent vectors of words P and sentences Q are estimated by minimizing the objective function:

$$\sum_i \sum_j W_{ij} \left(P_{\cdot,i} \cdot Q_{\cdot,j} - X_{ij}\right)^2 + \lambda ||P||_2^2 + \lambda ||Q||_2^2$$
$$\text{where } W_{i,j} = \begin{cases} 1, & \text{if } X_{ij} \neq 0 \\ w_m, & \text{if } X_{ij} = 0 \end{cases} \quad (1)$$

Equation 1 explicitly requires the latent vector of sentence $Q_{\cdot,j}$ to be not related to missing words ($P_{\cdot,i} \cdot Q_{\cdot,j}$ should be close to 0 for missing words $X_{ij} = 0$). Also weight w_m for missing words is very small to make sure latent vectors such as v_3 in Table 1 will not be chosen. In experiments we set $w_m = 0.01$. We refer to our approach as Weighted Textual Matrix Factorization (WTMF).

After we run WTMF on the sentence corpus, the similarity of the two sentences s_j and s_k can be computed by the inner product of $Q_{\cdot,j}$ and $Q_{\cdot,k}$.

3.3 Inference

The latent vectors in P and Q are first randomly initialized, then can be computed iteratively by the following equations (derivation is omitted due to limited space, but can be found in (Srebro and Jaakkola, 2003)):

$$P_{\cdot,i} = \left(Q\tilde{W}^{(i)}Q^\top + \lambda I\right)^{-1} Q\tilde{W}^{(i)}X_{i,\cdot}^\top$$
$$Q_{\cdot,j} = \left(P\tilde{W}^{(j)}P^\top + \lambda I\right)^{-1} P\tilde{W}^{(i)}X_{\cdot,j} \quad (2)$$

where $\tilde{W}^{(i)} = \text{diag}(W_{\cdot,i})$ is an $M \times M$ diagonal matrix containing ith row of weight matrix W. Similarly, $\tilde{W}^{(j)} = \text{diag}(W_{\cdot,j})$ is an $N \times N$ diagonal matrix containing jth column of W.

Since most of the cells have the same value of 0, the inference can be further optimized to save computation, which has been described in (Steck, 2010).

4 Data Preprocessing

The data sets for WTMF comprises two dictionaries WordNet (Fellbaum, 1998), Wiktionary,[2] and the Brown corpus. We did not link the senses between WordNet and Wiktionary, therefore the definition sentences are simply treated as individual documents. We crawl Wiktionary and remove the entries that are not tagged as noun, verb, adjective, or adverb, resulting in 220,000 entries. For both WordNet and Wiktionary, target words are added to the definition (e.g. the word $bank$ is added into the definition sentence of $bank\#n\#1$). Also usage examples are appended to definition sentences (hence sentences become short texts). For the Brown corpus, each sentence is treated as a document in order to create more co-occurrence. The importance of words in a sentence is estimated by the TF-IDF schema.

All data is tokenized, pos-tagged[3], and lemmatized[4]. To reduce word sparsity issue, we take an additional preprocessing step: for each lemmatized word, we find all its possible lemmas, and choose the most frequent lemma according to WordNet::QueryData. For example, the word *thinkings* is first lemmatized as *thinking*, then we discover *thinking* has possible lemmas *thinking* and *think*, finally we choose *think* as targeted lemma. The STS data is also preprocessed using the same pipeline.

5 Experiments

5.1 Setting

STS data: The sentence pair data in the STS task is collected from five sources: 1. MSR Paraphrase corpus (Dolan et al., 2004), 2. MSR video data (Chen and Dolan, 2011), 3. SMT europarl data,

[2]http://en.wiktionary.org/wiki/Wiktionary:Main_Page
[3]http://nlp.stanford.edu/software/tagger.shtml
[4]http://wn-similarity.sourceforge.net, WordNet::QueryData

models	MSRpar	MSRvid	SMT-eur	ON-WN	SMT-news
LDA	0.274	0.7682	0.452	0.619	0.366
WTMF	0.411(67/89)	0.835(11/89)	0.513(10/89)	0.727(1/89)	0.438(28/89)

Table 2: Performance of LDA and WTMF on each individual test set of Task 6 STS data

ALL	ALLnrm	Mean
0.695(20/89)	0.830(10/89)	0.608(19/89)

Table 3: Performance of WTMF on all test sets

4. OntoNotes-WordNet data (Hovy et al., 2006), 5. SMT news data.

Evaluation Metrics: Since the systems are required to assigned a similarity score to each sentence pair, Pearson's correlation is used to measure the performance of systems on each of the 5 data sets. However, measuring the overall performance on the concatenation of 5 data sets is rarely discussed in previous work. Accordingly the organizers of STS task provide three evaluation metrics: 1. ALL: Pearson correlation with the gold standard for the combined 5 data sets. 2. ALLnrm: Pearson correlation after the system outputs for each data set are fitted to the gold standard using least squares. 3. Mean: Weighted mean across the 5 data sets, where the weight depends on the number of pairs in the dataset.

WTMF Model: Our model is built on WordNet+Wiktionary+Brown+training data of STS. Each sentence of STS test data is transformed into a latent vector using Equation 2. Then sentence pair similarity is computed by the cosine similarity of the two latent vectors. We employ the parameters used in (Guo and Diab, 2012b) ($\lambda = 20, w_m = 0.01$).

5.2 Results

Table 3 summarizes the overall performance of WTMF on the concatenation of 5 data sets followed by the corresponding rank among all participating systems.[5] There are 88 submitted results in total and 1 baseline which is simply the cosine similarity of surface word vectors.

Table 2 compares the individual performance of LDA (trained on the same corpus) and WTMF on each data set. WTMF outperforms LDA by a large margin. This is because LDA only uses 10 observed words to infer a 100 dimension vector, while WTMF takes advantage of much more missing words to learn more robust latent semantic vectors.

WTMF model achieves great overall performance, with ranks 20, 10, 19 out of 89 reported results in three evaluation metrics respectively. It is worth noting that WTMF is unsupervised in that it does not use the training data similarity values, also the only feature WTMF uses is bag-of-words features without other information such as syntax, sentiment, etc. indicating that these additional features could lead to even more improvement.

Observing the individual performance on each of the 5 data set, we find WTMF ranks relatively high in the four data sets: MSRvid (11/89), SMT-eur (11/89), ON-WN (1/89), SMT-news (28/89). However, WTMF is outperformed by most of the systems on MSRpar data set (67/89). We analyze the data set and find that different from the other four data sets, MSRpar is related to a lot of other NLP topics such as textual entailment or sentiment coherence. Therefore, our feature set (bag of words) is too shallow for this data set indicating that using syntax and more semantically oriented features could be helpful.

6 Conclusions

We introduce a new latent variable model WTMF that is competitive with high dimensional approaches to the STS task. In WTMF model, we explicitly model missing words to alleviate the sparsity problem in modeling short texts. For future work, we would like to combine our methods with existing word similarity based approaches and add more nuanced features incorporating syntax and semantics in the latent model.

Acknowledgments

This research was funded by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), through the U.S. Army Research Lab. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of IARPA, the ODNI or the U.S. Government.

[5] http://www.cs.york.ac.uk/semeval-2012/task6/index.php?id=results-update

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), in conjunction with the First Joint Conference on Lexical and Computational Semantics (*SEM 2012)*.

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 805–810.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3.

David L. Chen and William B. Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

William Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics*.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Jin Feng, Yi-Ming Zhou, and Trevor Martin. 2008. Sentence similarity based on relevance. In *Proceedings of IPMU*.

Weiwei Guo and Mona Diab. 2012a. Learning the latent semantics of a concept from its definition. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.

Weiwei Guo and Mona Diab. 2012b. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.

Chukfong Ho, Masrah Azrifah Azmi Murad, Rabiah Abdul Kadir, and Shyamala C. Doraisamy. 2010. Word sense disambiguation-based sentence similarity. In *Proceedings of the 23rd International Conference on Computational Linguistics*.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*.

Aminul Islam and Diana Inkpen. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data*, 2.

David Kauchak and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*.

Thomas K Landauer, Peter W. Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse Processes*, 25.

Mirella Lapata and Regina Barzilay. 2005. Automatic evaluation of text coherence: Models and representations. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*.

Yuhua Li, Davi d McLean, Zuhair A. Bandar, James D. O Shea, and Keeley Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transaction on Knowledge and Data Engineering*, 18.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Articial Intelligence*.

James O'Shea, Zuhair Bandar, Keeley Crockett, and David McLean. 2008. A comparative study of two short text semantic similarity measures. In *Proceedings of the Agent and Multi-Agent Systems: Technologies and Applications, Second KES International Symposium (KES-AMSTA)*.

Daniel Ramage, Susan Dumais, and Dan Liebling. 2010. Characterizing microblogs with topic models. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*.

Nathan Srebro and Tommi Jaakkola. 2003. Weighted low-rank approximations. In *Proceedings of the Twentieth International Conference on Machine Learning*.

Harald Steck. 2010. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

George Tsatsaronis, Iraklis Varlamis, and Michalis Vazirgiannis. 2010. Text relatedness based on a word thesaurus. *Journal of Articial Intelligence Research*, 37.

Liang Zhou, Chin-Yew Lin, Dragos Stefan Munteanu, and Eduard Hovy. 2006. Paraeval: Using paraphrases to evaluate summaries automatically. In *Proceedings of Human Language Tech-nology Conference of the North American Chapter of the ACL,*.