Discriminative Improvements to Distributional Sentence Similarity

Yangfeng Ji
School of Interactive Computing
Georgia Institute of Technology
jiyfeng@gatech.edu

Jacob Eisenstein
School of Interactive Computing
Georgia Institute of Technology
jacobe@gatech.edu

Abstract

Matrix and tensor factorization have been applied to a number of semantic relatedness tasks, including paraphrase identification. The key idea is that similarity in the latent space implies semantic relatedness. We describe three ways in which labeled data can improve the accuracy of these approaches on paraphrase classification. First, we design a new discriminative term-weighting metric called TF-KLD, which outperforms TF-IDF. Next, we show that using the latent representation from matrix factorization as features in a classification algorithm substantially improves accuracy. Finally, we combine latent features with fine-grained n-gram overlap features, yielding performance that is 3% more accurate than the prior state-of-the-art.

1 Introduction

Measuring the semantic similarity of short units of text is fundamental to many natural language processing tasks, from evaluating machine translation (Kauchak and Barzilay, 2006) to grouping redundant event mentions in social media (Petrović et al., 2010). The task is challenging because of the infinitely diverse set of possible linguistic realizations for any idea (Bhagat and Hovy, 2013), and because of the short length of individual sentences, which means that standard bag-of-words representations will be hopelessly sparse.

Distributional methods address this problem by transforming the high-dimensional bag-of-words representation into a lower-dimensional latent space. This can be accomplished by factoring a matrix or tensor of term-context counts (Turney and Pantel, 2010); proximity in the induced latent space has been shown to correlate with semantic similarity (Mihalcea et al., 2006). However, factoring the term-context matrix means throwing away a considerable amount of information, as the original matrix of size $M \times N$ (number of instances by number of features) is factored into two smaller matrices of size $M \times K$ and $N \times K$, with $K \ll M, N$. If the factorization does not take into account labeled data about semantic similarity, important information can be lost.

In this paper, we show how labeled data can considerably improve distributional methods for measuring semantic similarity. First, we develop a new discriminative term-weighting metric called TF-KLD, which is applied to the term-context matrix *before* factorization. On a standard paraphrase identification task (Dolan et al., 2004), this method improves on both traditional TF-IDF and Weighted Textual Matrix Factorization (WTMF; Guo and Diab, 2012). Next, we convert the latent representations of each sentence pair into a feature vector, which is used as input to a linear SVM classifier. This yields further improvements and substantially outperforms the current state-of-the-art on paraphrase classification. We then add "fine-grained" features about the lexical similarity of the sentence pair. The combination of latent and fine-grained features yields further improvements in accuracy, demonstrating that these feature sets provide complementary information on semantic similarity.

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 891–896,
Seattle, Washington, USA, 18-21 October 2013.

2 Related Work

Without attempting to do justice to the entire literature on paraphrase identification, we note three high-level approaches: (1) string similarity metrics such as n-gram overlap and BLEU score (Wan et al., 2006; Madnani et al., 2012), as well as string kernels (Bu et al., 2012); (2) syntactic operations on the parse structure (Wu, 2005; Das and Smith, 2009); and (3) distributional methods, such as latent semantic analysis (LSA; Landauer et al., 1998), which are most relevant to our work. One application of distributional techniques is to replace individual words with distributionally similar alternatives (Kauchak and Barzilay, 2006). Alternatively, Blacoe and Lapata (2012) show that latent word representations can be combined with simple element-wise operations to identify the semantic similarity of larger units of text. Socher et al. (2011) propose a syntactically-informed approach to combine word representations, using a recursive auto-encoder to propagate meaning through the parse tree.

We take a different approach: rather than representing the meanings of individual words, we directly obtain a distributional representation for the entire sentence. This is inspired by Mihalcea et al. (2006) and Guo and Diab (2012), who treat sentences as pseudo-documents in an LSA framework, and identify paraphrases using similarity in the latent space. We show that the performance of such techniques can be improved dramatically by using supervised information to (1) reweight the individual distributional features and (2) learn the importance of each latent dimension.

3 Discriminative feature weighting

Distributional representations (Turney and Pantel, 2010) can be induced from a co-occurrence matrix $\mathbf{W} \in \mathbb{R}^{M \times N}$, where M is the number of instances and N is the number of distributional features. For paraphrase identification, each instance is a sentence; features may be unigrams, or may include higher-order n-grams or dependency pairs. By decomposing the matrix $\mathbf{W}$, we hope to obtain a latent representation in which semantically-related sentences are similar. Singular value decomposition (SVD) is traditionally used to perform this factorization. However, recent work has demonstrated the robustness of nonnegative matrix factorization (NMF; Lee and Seung, 2001) for text mining tasks (Xu et al., 2003; Arora et al., 2012); the difference from SVD is the addition of a non-negativity constraint in the latent representation based on non-orthogonal basis.

While $\mathbf{W}$ may simply contain counts of distributional features, prior work has demonstrated the utility of reweighting these counts (Turney and Pantel, 2010). TF-IDF is a standard approach, as the inverse document frequency (IDF) term increases the importance of rare words, which may be more discriminative. Guo and Diab (2012) show that applying a special weight to unseen words can further improvement performance on paraphrase identification.

We present a new weighting scheme, TF-KLD, based on supervised information. The key idea is to increase the weights of distributional features that are discriminative, and to decrease the weights of features that are not. Conceptually, this is similar to Linear Discriminant Analysis, a supervised feature weighting scheme for continuous data (Murphy, 2012).

More formally, we assume labeled sentence pairs of the form $\langle \vec{w}_i^{(1)}, \vec{w}_i^{(2)}, r_i \rangle$, where $\vec{w}_i^{(1)}$ is the binarized vector of distributional features for the first sentence, $\vec{w}_i^{(2)}$ is the binarized vector of distributional features for the second sentence, and $r_i \in \{0, 1\}$ indicates whether they are labeled as a paraphrase pair. Assuming the order of the sentences within the pair is irrelevant, then for k-th distributional feature, we define two Bernoulli distributions:

- $p_k = P(w_{ik}^{(1)} | w_{ik}^{(2)} = 1, r_i = 1)$. This is the probability that sentence $w_i^{(1)}$ contains feature k, given that k appears in $w_i^{(2)}$ and the two sentences are labeled as paraphrases, $r_i = 1$.

- $q_k = P(w_{ik}^{(1)} | w_{ik}^{(2)} = 1, r_i = 0)$. This is the probability that sentence $w_i^{(1)}$ contains feature k, given that k appears in $w_i^{(2)}$ and the two sentences are labeled as not paraphrases, $r_i = 0$.

The Kullback-Leibler divergence $KL(p_k || q_k) = \sum_x p_k(x) \log \frac{p_k(x)}{q_k(x)}$ is then a measure of the discriminability of feature k, and is guaranteed to be non-

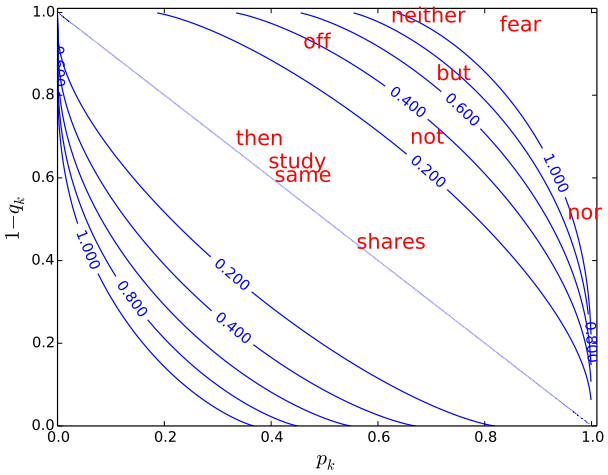

Figure 1: Conditional probabilities for a few hand-selected unigram features, with lines showing contours with identical KL-divergence. The probabilities are estimated based on the MSRPC training set (Dolan et al., 2004).

negative.[1] We use this divergence to reweight the features in $\mathbf{W}$ before performing the matrix factorization. This has the effect of increasing the weights of features whose likelihood of appearing in a pair of sentences is strongly influenced by the paraphrase relationship between the two sentences. On the other hand, if $p_k = q_k$, then the KL-divergence will be zero, and the feature will be ignored in the matrix factorization. We name this weighting scheme TF-KLD, since it includes the term frequency and the KL-divergence.

Taking the unigram feature *not* as an example, we have $p_k = [0.66, 0.34]$ and $q_k = [0.31, 0.69]$, for a KL-divergence of 0.25: the likelihood of this word being shared between two sentence is strongly dependent on whether the sentences are paraphrases. In contrast, the feature *then* has $p_k = [0.33, 0.67]$ and $q_k = [0.32, 0.68]$, for a KL-divergence of 3.9×10^{-4}. Figure 1 shows the distributions of these and other unigram features with respect to p_k and $1 - q_k$. The diagonal line running through the middle of the plot indicates zero KL-divergence, so features on this line will be ignored.

[1] We obtain very similar results with the opposite divergence $KL(q_k||p_k)$. However, the symmetric Jensen-Shannon divergence performs poorly.

1	unigram recall
2	unigram precision
3	bigram recall
4	bigram precision
5	dependency relation recall
6	dependency relation precision
7	BLEU recall
8	BLEU precision
9	Difference of sentence length
10	Tree-editing distance

Table 1: Fine-grained features for paraphrase classification, selected from prior work (Wan et al., 2006).

4 Supervised classification

While previous work has performed paraphrase classification using distance or similarity in the latent space (Guo and Diab, 2012; Socher et al., 2011), more direct supervision can be applied. Specifically, we convert the latent representations of a pair of sentences $\vec{v}_1$ and $\vec{v}_2$ into a sample vector,

$$\vec{s}(\vec{v}_1, \vec{v}_2) = \left[\vec{v}_1 + \vec{v}_2, |\vec{v}_1 - \vec{v}_2|\right], \qquad (1)$$

concatenating the element-wise sum $\vec{v}_1 + \vec{v}_2$ and absolute difference $|\vec{v}_1 - \vec{v}_2|$. Note that $\vec{s}(\cdot, \cdot)$ is symmetric, since $\vec{s}(\vec{v}_1, \vec{v}_2) = \vec{s}(\vec{v}_2, \vec{v}_1)$. Given this representation, we can use any supervised classification algorithm.

A further advantage of treating paraphrase as a supervised classification problem is that we can apply additional features besides the latent representation. We consider a subset of features identified by Wan et al. (2006), listed in Table 1. These features mainly capture fine-grained similarity between sentences, for example by counting specific unigram and bigram overlap.

5 Experiments

Our experiments test the utility of the TF-KLD weighting towards paraphrase classification, using the Microsoft Research Paraphrase Corpus (Dolan et al., 2004). The training set contains 2753 true paraphrase pairs and 1323 false paraphrase pairs; the test set contains 1147 and 578 pairs, respectively.

The TF-KLD weights are constructed from only the training set, while matrix factorizations are per-

formed on the entire corpus. Matrix factorization on both training and (unlabeled) test data can be viewed as a form of *transductive learning* (Gammerman et al., 1998), where we assume access to unlabeled test set instances.[2] We also consider an inductive setting, where we construct the basis of the latent space from only the training set, and then project the test set onto this basis to find the corresponding latent representation. The performance differences between the transductive and inductive settings were generally between 0.5% and 1%, as noted in detail below. We reiterate that the TF-KLD weights are never computed from test set data.

Prior work on this dataset is described in section 2. To our knowledge, the current state-of-the-art is a supervised system that combines several machine translation metrics (Madnani et al., 2012), but we also compare with state-of-the-art unsupervised matrix factorization work (Guo and Diab, 2012).

5.1 Similarity-based classification

In the first experiment, we predict whether a pair of sentences is a paraphrase by measuring their cosine similarity in latent space, using a threshold for the classification boundary. As in prior work (Guo and Diab, 2012), the threshold is tuned on held-out training data. We consider two distributional feature sets: **FEAT**$_1$, which includes unigrams; and **FEAT**$_2$, which also includes bigrams and unlabeled dependency pairs obtained from MaltParser (Nivre et al., 2007). To compare with Guo and Diab (2012), we set the latent dimensionality to $K = 100$, which was the same in their paper. Both SVD and NMF factorization are evaluated; in both cases, we minimize the Frobenius norm of the reconstruction error.

Table 2 compares the accuracy of a number of different configurations. The transductive TF-KLD weighting yields the best overall accuracy, achieving 72.75% when combined with non-negative matrix factorization. While NMF performs slightly better than SVD in both comparisons, the major difference is the performance of discriminative TF-KLD weighting, which outperforms TF-IDF regardless of the factorization technique. When we perform the matrix factorization on only the training data, the accuracy on the test set is 73.58%, with F1 score 80.55%.

[2]Another example of transductive learning in NLP is when Turian et al. (2010) induced word representations from a corpus that included both training and test data for their downstream named entity recognition task.

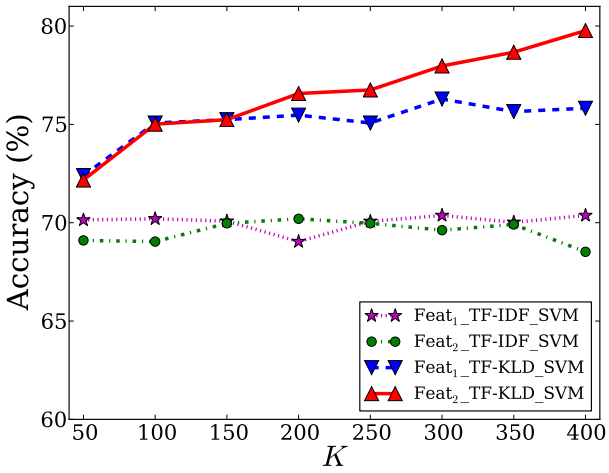

Figure 2: Accuracy of feature and weighting combinations in the classification framework.

5.2 Supervised classification

Next, we apply supervised classification, constructing sample vectors from the latent representation as shown in Equation 1. For classification, we choose a Support Vector Machine with a linear kernel (Fan et al., 2008), leaving a thorough comparison of classifiers for future work. The classifier parameter C is tuned on a development set comprising 20% of the original training set.

Figure 2 presents results for a range of latent dimensionalities. Supervised learning identifies the important dimensions in the latent space, yielding significantly better performance that the similarity-based classification from the previous experiment. In Table 3, we compare against prior published work, using the held-out development set to select the best value of K (again, $K = 400$). The best result is from TF-KLD, with distributional features FEAT$_2$, achieving 79.76% accuracy and 85.87% F1. This is well beyond all known prior results on this task. When we induce the latent basis from only the training data, we get 78.55% on accuracy and 84.59% F1, also better than the previous state-of-art.

Finally, we augment the distributional representation, concatenating the ten "fine-grained" features in Table 1 to the sample vectors described in Equation 1. As shown in Table 3, the accu-

Factorization	Feature set	Weighting	K	Measure	Accuracy (%)	F1
SVD	unigrams	TF-IDF	100	cosine sim.	68.92	80.33
NMF	unigrams	TF-IDF	100	cosine sim.	68.96	80.14
WTMF	unigrams	TF-IDF	100	cosine sim.	71.51	not reported
SVD	unigrams	TF-KLD	100	cosine sim.	72.23	81.19
NMF	unigrams	TF-KLD	100	cosine sim.	**72.75**	**81.48**

Table 2: Similarity-based paraphrase identification accuracy. Results for WTMF are reprinted from the paper by Guo and Diab (2012).

	Acc.	F1
Most common class	66.5	79.9
(Wan et al., 2006)	75.6	83.0
(Das and Smith, 2009)	73.9	82.3
(Das and Smith, 2009) with 18 features	76.1	82.7
(Bu et al., 2012)	76.3	not reported
(Socher et al., 2011)	76.8	83.6
(Madnani et al., 2012)	77.4	84.1
FEAT_2, TF-KLD, SVM	79.76	85.87
FEAT_2, TF-KLD, SVM, Fine-grained features	**80.41**	**85.96**

Table 3: Supervised classification. Results from prior work are reprinted.

racy now improves to 80.41%, with an F1 score of 85.96%. When the latent representation is induced from only the training data, the corresponding results are 79.94% on accuracy and 85.36% F1, again better than the previous state-of-the-art. These results show that the information captured by the distributional representation can still be augmented by more fine-grained traditional features.

6 Conclusion

We have presented three ways in which labeled data can improve distributional measures of semantic similarity at the sentence level. The main innovation is TF-KLD, which discriminatively reweights the distributional features *before* factorization, so that discriminability impacts the induction of the latent representation. We then transform the latent representation into a sample vector for supervised learning, obtaining results that strongly outperform the prior state-of-the-art; adding fine-grained lexical features further increases performance. These ideas may have applicability in other semantic similarity tasks, and we are also eager to apply them to new, large-scale automatically-induced paraphrase corpora (Ganitkevitch et al., 2013).

Acknowledgments

We thank the reviewers for their helpful feedback, and Weiwei Guo for quickly answering questions about his implementation. This research was supported by a Google Faculty Research Award to the second author.

References

Sanjeev Arora, Rong Ge, and Ankur Moitra. 2012. Learning Topic Models - Going beyond SVD. In *FOCS*, pages 1–10.

Rahul Bhagat and Eduard Hovy. 2013. What Is a Paraphrase? *Computational Linguistics*.

William Blacoe and Mirella Lapata. 2012. A Comparison of Vector-based Representations for Semantic Composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 546–556, Stroudsburg, PA, USA. Association for Computational Linguistics.

Fan Bu, Hang Li, and Xiaoyan Zhu. 2012. String Rewriting kernel. In *Proceedings of ACL*, pages 449–458. Association for Computational Linguistics.

Dipanjan Das and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the Joint Conference*

of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing, pages 468–476, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *COLING*.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9:1871–1874.

Alexander Gammerman, Volodya Vovk, and Vladimir Vapnik. 1998. Learning by transduction. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 148–155. Morgan Kaufmann Publishers Inc.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of NAACL*, pages 758–764. Association for Computational Linguistics.

Weiwei Guo and Mona Diab. 2012. Modeling Sentences in the Latent Space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 864–872, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Kauchak and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proceedings of NAACL*, pages 455–462. Association for Computational Linguistics.

Thomas Landauer, Peter W. Foltz, and Darrel Laham. 1998. Introduction to Latent Semantic Analysis. *Discource Processes*, 25:259–284.

Daniel D. Lee and H. Sebastian Seung. 2001. Algorithms for Non-Negative Matrix Factorization. In *Advances in Neural Information Processing Systems (NIPS)*.

Nitin Madnani, Joel R. Tetreault, and Martin Chodorow. 2012. Re-examining Machine Translation Metrics for Paraphrase Identification. In *HLT-NAACL*, pages 182–190. The Association for Computational Linguistics.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*.

Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Saša Petrović, Miles Osborne, and Victor Lavrenko. 2010. Streaming first story detection with application to twitter. In *Proceedings of HLT-NAACL*, pages 181–189. Association for Computational Linguistics.

Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011. Dynamic Pooling And Unfolding Recursive Autoencoders For Paraphrase Detection. In *Advances in Neural Information Processing Systems (NIPS)*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word Representation: A Simple and General Method for Semi-Supervised Learning. In *ACL*, pages 384–394.

Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *JAIR*, 37:141–188.

Ssephen Wan, Mark Dras, Robert Dale, and Cecile Paris. 2006. Using Dependency-based Features to Take the "Para-farce" out of Paraphrase. In *Proceedings of the Australasian Language Technology Workshop*.

Dekai Wu. 2005. Recognizing paraphrases and textual entailment using inversion transduction grammars. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 25–30. Association for Computational Linguistics.

Wei Xu, Xin Liu, and Yihong Gong. 2003. Document Clustering based on Non-Negative Matrix Factorization. In *SIGIR*, pages 267–273.