Metric Learning for Synonym Acquisition

Nobuyuki Shimizu
Information Technology Center
University of Tokyo
shimizu@r.dl.itc.u-tokyo.ac.jp

Masato Hagiwara
Graduate School of Information Science
Nagoya University
hagiwara@kl.i.is.nagoya-u.ac.jp

Yasuhiro Ogawa and **Katsuhiko Toyama**
Graduate School of Information Science
Nagoya University
{yasuhiro,toyama}@kl.i.is.nagoya-u.ac.jp

Hiroshi Nakagawa
Information Technology Center
University of Tokyo
n3@dl.itc.u-tokyo.ac.jp

Abstract

The distance or similarity metric plays an important role in many natural language processing (NLP) tasks. Previous studies have demonstrated the effectiveness of a number of metrics such as the Jaccard coefficient, especially in synonym acquisition. While the existing metrics perform quite well, to further improve performance, we propose the use of a supervised machine learning algorithm that fine-tunes them. Given the known instances of similar or dissimilar words, we estimated the parameters of the Mahalanobis distance. We compared a number of metrics in our experiments, and the results show that the proposed metric has a higher mean average precision than other metrics.

1 Introduction

Accurately estimating the semantic distance between words in context has applications for machine translation, information retrieval (IR), speech recognition, and text categorization (Budanitsky and Hirst, 2006), and it is becoming clear that a combination of corpus statistics can be used with a dictionary, thesaurus, or other knowledge source such as WordNet or Wikipedia, to increase the accuracy of semantic distance estimation (Mohammad and Hirst, 2006). Although compiling such resources is labor intensive and achieving wide coverage is difficult, these resources to some extent explicitly capture semantic structures of concepts and words. In contrast, corpus statistics achieve wide coverage, but the semantic structure of a concept is only implicitly represented in the context. Assuming that two words are semantically closer if they occur in similar contexts, statistics on the contexts of words can be gathered and compared for similarity, by using a metric such as the Jaccard coefficient.

Our proposal is to extend and fine-tune the latter approach with the training data obtained from the former. We apply metric learning to this task. Although still in their infancy, distance metric learning methods have undergone rapid development in the field of machine learning. In a setting similar to semi-supervised clustering, where known instances of similar or dissimilar objects are given, a metric such as the Mahalanobis distance can be learned from a few data points and tailored to fit a particular purpose. Although classification methods such as logistic regression now play important roles in natural language processing, the use of metric learning has yet to be explored.

Since popular current methods for synonym acquisition require no statistical learning, it seems that supervised machine learning should easily outperform them. Unfortunately, there are obstacles to overcome. Since metric learning algorithms usually learn the parameters of a Mahalanobis distance, the number of parameters is quadratic to the number of features. They learn how two features should interact to produce the final metric. While traditional metrics forgo examining of the interactions entirely, in applying metrics such as Jaccard coefficient, it is not uncommon nowadays to use more than 10,000 features, a number that a typical metric learner is incapable of processing. Thus we have two options: one is to find the most important features and model the interactions between

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 793–800
Manchester, August 2008

them, and the other is simply to use a large number of features. We experimentally examined the two options and found that metric learning is useful in synonym acquisition, despite it utilizing fewer features than traditional methods.

The remainder of this paper is organized as follows: in section 2, we review prior work on synonym acquisition and metric learning. In section 3, we introduce the Mahalanobis distance metric and a learning algorithm based on this metric. In section 4 and 5, we explain the experimental settings and propose the use of normalization to make the Mahalanobis distances work in practice, and then in section 6, we discuss issues we encountered when applying this metric to synonym acquisition. We conclude in section 7.

2 Prior Work

As this paper is based on two different lines of research, we first review the work in synonym acquisition, and then review the work in generic metric learning. To the best of the authors' knowledge, none of the metric learning algorithms have been applied to automatic synonym acquisition.

Synonym relation is important lexical knowledge for many natural language processing tasks including automatic thesaurus construction (Croach and Yang, 1992; Grefenstette, 1994) and IR (Jing and Croft, 1994). Various methods (Hindle, 1990; Lin, 1998) of automatically acquiring synonyms have been proposed. They are usually based on the distributional hypothesis (Harris, 1985), which states that semantically similar words share similar contexts, and they can be roughly viewed as the combinations of two steps: context extraction and similarity calculation. The former extracts useful features from the contexts of words, such as surrounding words or dependency structure. The latter calculates how semantically similar two given words are based on similarity or distance metrics.

Many studies (Lee, 1999; Curran and Moens, 2002; Weeds et al., 2004) have investigated similarity calculation, and a variety of distance/similarity measures have already been compared and discussed. Weeds et al.'s work is especially useful because it investigated the characteristics of metrics based on a few criteria such as the relative frequency of acquired synonyms and clarified the correlation between word frequency, distributional generality, and semantic generality. However, all of the existing research conducted only a posteriori comparison, and as Weeds et al. pointed out, there is no one best measure for all applications. Therefore, the metrics must be tailored to applications, even to corpora and other settings.

We next review the prior work in generic metric learning. Most previous metric learning methods learn the parameters of the Mahalanobis distance. Although the algorithms proposed in earlier work (Xing et al., 2002; Weinberger et al., 2005; Globerson and Roweis, 2005) were shown to yield excellent classification performance, these algorithms all have worse than cubic computational complexity in the dimensionality of the data. Because of the high dimensionality of our objects, we opted for information-theoretic metric learning proposed by (Davis et al., 2007). This algorithm only uses an operation quadratic in the dimensionality of the data.

Other work on learning Mahalanobis metrics includes online metric learning (Shalev-Shwartz et al., 2004), locally-adaptive discriminative methods (Hastie and Tibshirani, 1996), and learning from relative comparisons (Schutz and Joahims, 2003). Non-Mahalanobis-based metric learning methods have also been proposed, though they seem to suffer from suboptimal performance, non-convexity, or computational complexity. Examples include neighborhood component analysis (Goldberger et al., 2004).

3 Metric Learning

3.1 Problem Formulation

To set the context for metric learning, we first describe the objects whose distances from one another we would like to know. As noted above regarding the distributional hypothesis, our object is the context of a target word. To represent the context, we use a sparse vector in R^d. Each dimension of an input vector represents a feature of the context, and its value corresponds to the strength of the association. The vectors of two target words represent their contexts as points in multidimensional feature-space. A suitable metric (for example, Euclidean) defines the distance between the two points, thereby estimating the semantic distance between the target words.

Given points $x_i, x_j \in R^d$, the (squared) Mahalanobis distance between them is parameterized by a positive definite matrix A as follows $d_A(x_i, x_j) = (x_i - x_j)^\top A(x_i - x_j)$. The Ma-

halanobis distance is a straightforward extension of the standard Euclidean distance. If we let A be the identity matrix, the Mahalanobis distance reduces to the Euclidean distance. Our objective is to obtain the positive definite matrix A that parameterizes the Mahalanobis distance, so that the distance between the vectors of two synonymous words is small, and the distance between the vectors of two dissimilar words is large. Stated more formally, the Mahalanobis distance between two similar points must be smaller than a given upper bound, i.e., $d_A(x_i, x_j) \leq u$ for a relatively small value of u. Similarly, two points are dissimilar if $d_A(x_i, x_j) \geq l$ for sufficiently large l.

As we discuss below, we were able to use the Euclidean distance to acquire synonyms quite well. Therefore, we would like the positive definite matrix A of the Mahalanobis distance to be close to the identity matrix I. This keeps the Mahalanobis distance similar to the Euclidean distance, which would help to prevent overfitting the data. To optimize the matrix, we follow the information theoretic metric learning approach described in (Davis et al., 2007). We summarize the problem formulation advocated by this approach in this section and the learning algorithm in the next section.

To define the closeness between A and I, we use a simple bijection (up to a scaling function) from the set of Mahalanobis distances to the set of equal mean multivariate Gaussian distributions. Without loss of generalization, let the equal mean be μ. Then given a Mahalanobis distance parameterized by A, the corresponding Gaussian is $p(x; A) = \frac{1}{Z}\exp(-\frac{1}{2}d_A(x, \mu))$ where Z is the normalizing factor. This enables us to measure the distance between two Mahalanobis distances with the Kullback-Leibler (KL) divergence of two Gaussians:

$$KL(p(x; I)||p(x; A)) = \int p(x, I) \log\left(\frac{p(x; I)}{p(x; A)}\right) dx.$$

Given pairs of similar points S and pairs of dissimilar points D, the optimization problem is:

$$\begin{array}{rll} \min_A & KL(p(x; I)||p(x; A)) & \\ \text{subject to} & d_A(x_i, x_j) \leq u & (i, j) \in S \\ & d_A(x_i, x_j) \geq l & (i, j) \in D \end{array}$$

3.2 Learning Algorithm

(Davis and Dhillon, 2006) has shown that the KL divergence between two multivariate Gaussians can be expressed as the convex combination of a Mahalanobis distance between mean vectors and the LogDet divergence between the covariance matrices. The LogDet divergence equals

$$D_{ld}(A, A_0) = tr(AA_0^{-1}) - \log\det(AA_0^{-1}) - n$$

for n by n matrices A, A_0. If we assume the means of the Gaussians to be the same, we have

$$KL(p(x; A_0||p(x, A)) = \frac{1}{2}D_{ld}(A, A_0)$$

The optimization problem can be restated as

$$\begin{array}{rll} \min_{A\succeq 0} & D_{ld}(A, I) & \\ \text{s.t.} & tr(A(x_i - x_j)(x_i - x_j)^\top) \leq u & (i, j) \in S \\ & tr(A(x_i - x_j)(x_i - x_j)^\top) \geq l & (i, j) \in D \end{array}$$

We then incorporate slack variables into the formulation to guarantee the existence of a feasible solution for A. The optimization problem becomes:

$$\begin{array}{rll} \min_{A\succeq 0} & D_{ld}(A, I) + \gamma D_{ld}(diag(\xi), diag(\xi_0)) & \\ \text{s.t.} & tr(A(x_i - x_j)(x_i - x_j)^\top) \leq \xi_{c(i,j)} & (i, j) \in S \\ & tr(A(x_i - x_j)(x_i - x_j)^\top) \geq \xi_{c(i,j)} & (i, j) \in D \end{array}$$

where $c(i, j)$ is the index of the (i, j)-th constraint and ξ is a vector of slack variables whose components are initialized to u for similarity constraints and l for dissimilarity constraints. The tradeoff between satisfying the constraints and minimizing $D_{ld}(A, I)$ is controlled by the parameter γ. To solve this optimization problem, the algorithm shown in Algorithm 3.1 repeatedly projects the current solution onto a single constraint.

This completes the summary of (Davis et al., 2007).

4 Experimental Settings

In this section, we describe the experimental settings including the preprocessing of data and features, creation of the query word sets, and settings of the cross validation.

4.1 Features

We used a dependency structure as the context for words because it is the most widely used and one of the best performing contextual information in the past studies (Ruge, 1997; Lin, 1998). As the extraction of an accurate and comprehensive dependency structure is in itself a complicated task, the sophisticated parser RASP Toolkit 2 (Briscoe et al., 2006) was utilized to extract this kind of word relation.

Let $N(w, c)$ be the raw cooccurrence count of word w and context c, the grammatical relation

Algorithm
3.1: INFORMATION THEORETIC METRIC LEARNING

Input :
X(d by n matrix), I(identity matrix)
S(set of similar pairs), D(set of dissimilar pairs)
γ(slack parameter), c(constraint index function)
u, l(distance thresholds)
Output :
A(Mahalanobis matrix)

$A := I$
$\lambda_{ij} := 0$
$\xi_{c(i,j)} := u$ for $(i,j) \in S$; otherwise, $\xi_{c(i,j)} := l$
repeat
  Pick a constraint $(i,j) \in S$ or $(i,j) \in D$
  $p := (x_i - x_j)^\top A (x_i - x_j)$
  $\delta := 1$ if $(i,j) \in S, -1$ otherwise.
  $\alpha := \min(\lambda_{ij}, \frac{\delta}{2}(\frac{1}{p} - \frac{\gamma}{\xi_{c(i,j)}}))$
  $\beta := \delta\alpha/(1 - \delta\alpha\xi_{c(i,j)})$
  $\xi_{c(i,j)} := \gamma\xi_{c(i,j)}/(\gamma + \delta\alpha\xi_{c(i,j)})$
  $\lambda_{ij} := \lambda_{ij} - \alpha$
  $A := A + \beta A(x_i - x_j)(x_i - x_j)^\top A$
until convergence
return (A)

in which w occurs. These raw counts were obtained from New York Times articles (July 1994) extracted from English Gigaword [1]. The section consists of 7,593 documents and approx. 5 million words. As discussed below, we limited the vocabulary to the nouns in the Longman Defining Vocabulary (LDV) [2]. The features were constructed by weighting them using pointwise mutual information: $\mathrm{wgt}(w,c) = \mathrm{PMI}(w,c) = \log \frac{P(w,c)}{P(w)P(c)}$.

Co-occurrence data constructed this way can yield more than 10,000 context types, rendering metric learning impractical. As the applications of feature selection reduce the performance of the baseline metrics, we tested them in two different settings: with and without feature selection. To mitigate this problem, we applied a feature selection technique to reduce the feature dimensionality. We selected features using two approaches. The first approach is a simple frequency cutoff, applied as a pre-processing to filter out words and contexts with low frequency and to reduce computational cost. Specifically, all words w such that $\sum_c N(w,c) < \theta_f$ and contexts c such that $\sum_w N(w,c) < \theta_f$, with $\theta_f = 5$, are removed from the co-occurrence data.

The second approach is feature selection by context importance (Hagiwara et al., 2008). First, the *context importance* score for each context type is calculated, and then the least important context types are eliminated, until a desired numbers of them remains. To measure the *context importance* score, we used the number of unique words the context co-occurs with: $df(c) = |\{w|N(w,c) > 0\}|$. We adopted this context selection criterion on the assumption that the contexts shared by many words should be informative, and the synonym acquisition performance based on normal distributional similarity calculation retains its original level of performance until up to almost 90% of context types are eliminated (Hagiwara et al., 2008). In our experiment, we selected features rather aggressively, finally using only 10% of the original contexts. These feature reduction operations reduced the dimensionality to a figure as small as 1,281, while keeping the performance loss at a minimum.

4.2 Similarity and Distance Functions

We compared seven similarity/distance functions in our experiments: cosine similarity, Euclidean distance, Manhattan distance, Jaccard coefficient, vector-based Jaccard coefficient (Jaccardv), Jensen-Shannon Divergence (JS) and skew divergence (SD99). We first define some notations. Let $C(w)$ be the set of context types that co-occur with word w, i.e., $C(w) = \{c|N(w,c) > 0\}$, and $\mathbf{w}_i$ be the feature vector corresponding to word w, i.e., $\mathbf{w}_i = [\mathrm{wgt}(w_i, c_1) \ ... \ \mathrm{wgt}(w_i, c_M)]^\top$. The first three, the cosine, Euclidean and Manhattan distance, are vector-based metrics.

cosine similarity

$$\frac{\mathbf{w}_1 \cdot \mathbf{w}_2}{||\mathbf{w}_1|| \cdot ||\mathbf{w}_2||}$$

Euclidean distance

$$\sqrt{\sum_{c \in C(w_1) \cup C(w_2)} (\mathrm{wgt}(w_1,c) - \mathrm{wgt}(w_2,c))^2}$$

Manhattan distance

$$\sum_{c \in C(w_1) \cup C(w_2)} |\mathrm{wgt}(w_1,c) - \mathrm{wgt}(w_2,c)|$$

Jaccard coefficient

$$\frac{\sum_{c \in C(w_1) \cap C(w_2)} \min(\mathrm{wgt}(w_1,c), \mathrm{wgt}(w_2,c))}{\sum_{c \in C(w_1) \cup C(w_2)} \max(\mathrm{wgt}(w_1,c), \mathrm{wgt}(w_2,c))},$$

[1] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05

[2] http://www.cs.utexas.edu/users/kbarker/working_notes/ldoce-vocab.html

vector-based Jaccard coefficient (Jaccardv)

$$\frac{\mathbf{w}_i \cdot \mathbf{w}_j}{||\mathbf{w}_i|| + ||\mathbf{w}_j|| - \mathbf{w}_i \cdot \mathbf{w}_j}.$$

Jensen-Shannon divergence (JS)

$$\frac{1}{2}\{KL(p_1||m) + KL(p_2||m)\},\ m = p_1 + p_2.$$

JS and SD99 are based on the KL divergence, so the vectors must be normalized to form a probability distribution. For notational convenience, we let p_i be the probability distribution representation of feature vector $\mathbf{w}_i$, i.e., $p_i(c) = N(\mathbf{w}_i, c)/N(\mathbf{w}_i)$. While the KL divergence suffers from the so-called zero-frequency problem, a symmetric version of the KL divergence called the Jensen-Shannon divergence naturally avoids it.

skew divergence (SD99)

$$KL(p_1||\alpha p_2 + (1-\alpha)p_1).$$

As proposed by (Lee, 2001), the skew divergence also avoids the zero-frequency problem by mixing the original distribution with the target distribution. Parameter α is set to 0.99.

4.3 Query Word Set and Cross Validation

To formalize the experiments, we must prepare a set of query words for which synonyms are known in advance. We chose the Longman Defining Vocabulary (LDV) as the candidate set of query words. For each word in the LDV, we consulted three existing thesauri: Roget's Thesaurus (Roget, 1995), Collins COBUILD Thesaurus (Collins, 2002), and WordNet (Fellbaum, 1998). Each LDV word was looked up as a noun to obtain the union of synonyms. After removing words marked "idiom", "informal" or "slang" and phrases comprised of two or more words, this union was used as the reference set of query words. LDV words for which no noun synonyms were found in any of the reference thesauri were omitted. From the remaining 771 LDV words, there were 231 words that had five or more synonyms in the combined thesaurus. We selected these 231 words to be the query words and distributed them into five partitions so as to conduct five-fold cross validation. Four partitions were used in training, and the remaining partition was used in testing. For each fold, we created the training set from four partitions as follows; for each query word in the partitions, we randomly selected five synonymous words and added the pairs of query words and synonymous words to S, the set of similar pairs. Similarly, five pairs of query words and dissimilar words were randomly added to D, the set of dissimilar pairs. The training set for each fold consisted of S and D. Since a learner trained on an imbalanced dataset may not learn to discriminate enough between classes, we sampled dissimilar pairs to create an evenly distributed training dataset.

To make the evaluation realistic, we used a different method to create the test set: we paired each query word with each of the 771 remaining words to form the test set. Thus, in each fold, the training set had an equal number of positive and negative pairs, while in the test set, negative pairs outnumbered the positive pairs. While this is not a typical setting for cross validation, it renders the evaluation more realistic since an automatic synonym acquisition system in operation must be able to pick a few synonyms from a large number of dissimilar words.

The meta-parameters of the metric learning model were simply set $u = 1$, $l = 2$ and $\gamma = 1$. Each training set consisted of 1,850 pairs, and the test set consisted of 34,684 pairs. Since we conducted five-fold cross validation, the reported performance in this paper is actually a summary over different folds.

4.4 Evaluation Measures

We used an evaluation program for KDD Cup 2004 (Caruana et al., 2004) called Perf to measure the effectiveness of the metrics in acquiring synonyms. To use the program, we used the following formula to convert each distance metric to a similarity metric. $s(x_i, x_j) = 1/(1 + \exp(d(x_i, x_j)))$. Below, we summarize the three measures we used: Mean Average Precision, TOP1, and Average Rank of Last Synonym.

Mean Average Precision (APR)

Perf implements a definition of average precision sometimes called "expected precision". Perf calculates the precision at every recall where it is defined. For each of these recall values, Perf finds the threshold that produces the maximum precision, and takes the average over all of the recall values greater than 0. Average precision is measured on each query, and then the mean of each query's average precision is used as the final metric. A mean average precision of 1.0 indicates perfect prediction. The lowest possible mean average

precision is 0.0.

Average Rank of Last Synonym (RKL)

As in other evaluation measures, synonym candidates are sorted by predicted similarity, and this metric measures how far down the sorted cases we must go to find the last true synonym. A rank of 1 indicates that the last synonym is placed in the top position. Given a query word, the highest obtainable rank is N if there are N synonyms in the corpus. The lower this measure is the better. Average ranks near 771 indicate poor performance.

TOP1

In each query, synonym candidates are sorted by predicted similarity. If the word that ranks at the top (highest similarity to the query word) is a true synonym of the query word, Perf scores a 1 for that query, and 0 otherwise. If there are ties, Perf scores 0 unless all of the tied cases are synonyms. TOP1 score ranges from 1.0 to 0.0. To achieve 1.0, perfect TOP1 prediction, a similarity metric must place a true synonym at the top of the sorted list in every query. In the next section, we report the mean of each query's TOP1.

5 Results

The evaluations of the metrics are listed in Table 1. The figure on the left side of $\rightarrow$ represents the performance with 1,281 features, and that on the right side with 12,812 features. Of all the metrics in Table 1, only the Mahalanobis L2 is trained with the previously presented metric learning algorithm. Thus, the values for the Mahalanobis L2 are produced by the five-fold cross validation, while the rest are given by the straight application of the metrics discussed in Section 4.2 to the same dataset. Strictly speaking, this is not a fair comparison, since we ought to compare a supervised learning with a supervised learning. However, our baseline is not the simple Euclidean distance; it is the Jaccard coefficient and cosine similarity, a handcrafted, best performing metric for synonym acquisition, with 10 times as many features.

The computational resources required to obtain the Mahalanobis L2 results were as follows: in the training phase, each fold of cross validation took about 80 iterations (less than one week) to converge on a Xeon 5160 3.0GHz. The time required to use the learned distance was a few hours at most.

At first, we were unable to perform competitively with the Euclidean distance. As seen in Table 1, the TOP1 measure of the Euclidean distance is only 1.732%. This indicates that the likelihood of finding the first item on the ranked list to be a true synonym is 1.732%. The vector-based Jaccard coefficient performs much better than the Euclidean distance, placing a true synonym at the top of the list 30.736% of the time.

Table 2 shows the Top 10 Words for Query "branch". The results for the Euclidean distance rank "hut" and other dissimilar words highly. This is because the norm of such vectors is small, and in a high dimensional space, the sparse vectors near the origin are relatively close to many other sparse vectors. To overcome this problem, we normalized the input vectors by the L2 norm $x' = x/||x||$ This normalization enables the Euclidean distance to perform very much like the cosine similarity, since the Euclidean distance between points on a sphere acts like the angle between the vectors. Surprisingly, normalization by L2 did not affect other metrics all that much; while the performances of some metrics improved slightly, the L2 normalization lowered that of the Jaccardv metric.

Once we learned the normalization trick, the learned Mahalanobis distance consistently outperformed all other metrics, including the ones with 10 times more features, in all three evaluation measures, achieving an APR of 18.66%, RKL of 545.09 and TOP1 of 45.455%.

6 Discussion

Examining the learned Mahalanobis matrix revealed interesting features. The matrix essentially shows the covariance between features. While it was not as heavily weighted as the diagonal elements, we found that its positive non-diagonal elements were quite interesting. They indicate that some of the useful features for finding synonyms are correlated and somewhat interchangeable. The example includes a pair of features, (dobj begin *) and (dobj end *). It was a pleasant surprise to see that one implies the other. Among the diagonal elements of the matrix, one of the heaviest features was being the direct object of "by". This indicates that being the object of the preposition "by" is a good indicator that two words are similar. A closer inspection of the NYT corpus showed that this preposition overwhelmingly takes a person or organization as its object, indicating that words with this feature belong to the same class of a person or organization. Similarly, the class

Metric	APR	RKL	TOP1
Cosine	0.1184 → 0.1324	580.27 → 579.00	0.2987 → 0.3160
Euclidean	0.0229 → 0.0173	662.74 → 695.71	0.0173 → 0.0000
Euclidean L2	0.1182 → 0.1324	580.30 → 578.99	0.2943 → 0.3160
Jaccard	0.1120 → 0.1264	580.76 → 579.51	0.2684 → 0.2943
Jaccard L2	0.1113 → 0.1324	580.29 → 570.88	0.2640 → 0.2987
Jaccardv	0.1189 → 0.1318	580.50 → 580.19	0.3073 → 0.3030
Jaccardv L2	0.1184 → 0.1254	580.27 → 570.00	0.2987 → 0.3160
JS	0.0199 → 0.0170	681.97 → 700.53	0.0129 → 0.0000
JS L2	0.0229 → 0.0173	679.21 → 699.00	0.0303 → 0.0086
Manhattan	0.0181 → 0.0168	687.73 → 701.47	0.0043 → 0.0000
Manhattan L2	0.0185 → 0.0170	686.56 → 701.11	0.0043 → 0.0086
SD99	0.0324 → 0.1039	640.71 → 588.16	0.0173 → 0.2640
SD99 L2	0.0334 → 0.1117	633.32 → 586.78	0.0216 → 0.2900
Mahalanobis L2	**0.1866**	**545.09**	**0.4545**

Table 1: Evaluation of Various Metrics, as Number of Features Increase from 1,281 to 12,812

	Cosine	Euclidean	Euclidean L2	Jaccard	Jaccardv	Mahalanobis L2
1	(*) office	hut	(*) office	(*) office	(*) office	(*) division
2	area	wild	area	border	area	group
3	(*) division	polish	(*) division	area	(*) division	(*) office
4	border	thirst	border	plant	border	line
5	group	hollow	group	(*) division	group	period
6	organization	shout	organization	mouth	organization	organization
7	store	fold	store	store	store	(*) department
8	mouth	dear	mouth	circle	mouth	charge
9	plant	hate	plant	stop	plant	world
10	home	wake	home	track	home	body

(*) = a true synonym

Table 2: Top 10 Words for Query "branch"

of words that "to" and "within", take as an objects were clear from the corpus: "to" takes a person or place, "within" takes duration of time [3]. Other heavy features includes being the object of "write" or "about". While not obvious, we postulate that having these words as a part of the context indicates that a word is an event of some type.

7 Conclusion

We applied metric learning to automatic synonym acquisition for the first time, and our experiments showed that the learned metric significantly outperforms existing similarity metrics. This outcome indicates that while we must resort to feature selection to apply metric learning, the performance gain from the supervised learning is enough to offset the disadvantage and justify its usage in some applications. This leads us to think that a combination of the learned metric with unsupervised metrics with even more features may produces the best results. We also discussed interesting features found in the learned Mahalanobis matrix. Since metric learning is known to boost clustering performance in a semi-supervised clustering setting, we believe these automatically identified features would be helpful in assigning a target word to a word class.

[3]Interestingly, we note that not all prepositions were as heavy: "beyond" and "without" were relatively light among the diagonal elements. In the NYT corpus, the class of words they take was not as clear as, for example, "by".

References

T. Briscoe, J. Carroll and R. Watson. 2006. The Second Release of the RASP System. *Proc. of the COLING/ACL 2006 Interactive Presentation Sessions*, 77–80.

T. Briscoe, J. Carroll, J. Graham and A. Copestake, 2002. Relational evaluation schemes. *Proc. of the Beyond PARSEVAL Workshop at the Third International Conference on Language Resources and Evaluation*, 4–8.

A. Budanitsky and G. Hirst. 2006. Evaluating WordNet-based measures of semantic distance. *Computational Linguistics*, 32(1):13–47.

R. Caruana, T. Jachims and L. Backstrom. 2004. KDD-Cup 2004: results and analysis *ACM SIGKDD Explorations Newslatter*, 6(2):95–108.

C. J. Croach and B. Yang. 1992. Experiments in automatic statistical thesaurus construction. *the 15th*

Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 77–88.

J. R. Curran and M. Moens. 2002. Improvements in automatic thesaurus extraction. In Workshop on Unsupervised Lexical Acquisition. *Proc. of the ACL SIGLEX*, 231–238.

J. V. Davis and I. S. Dhillon. 2006. Differential Entropic Clustering of Multivariate Gaussians. *Advances in Neural Information Processing Systems (NIPS)*.

J. V. Davis, B. Kulis, P. Jain, S. Sra and I. S. Dhillon. 2007. Information Theoretic Metric Learning. *Proc. of the International Conference on Machine Learning (ICML)*.

A. Globerson and S. Roweis. 2005. Metric Learning by Collapsing Classes. *Advances in Neural Information Processing Systems (NIPS)*.

J. Goldberger, S. Roweis, G. Hinton and R. Salakhutdinov. 2004. Neighbourhood Component Analysis. *Advances in Neural Information Processing Systems (NIPS)*.

G. Grefenstette. 1994. Explorations in Automatic Thesuarus Discovery. Kluwer Academic Publisher.

M. Hagiwara, Y. Ogawa, and K. Toyama. 2008. Context Feature Selection for Distributional Similarity. *Proc. of IJCNLP-08*, 553–560.

Z. Harris. 1985. Distributional Structure. Jerrold J. Katz (ed.) *The Philosophy of Linguistics*. Oxford University Press. 26–47.

T. Hastie and R. Tibshirani. 1996. Discriminant adaptive nearest neighbor classification. *Pattern Analysis and Machine Intelligence, 18*, 607–616.

D. Hindle. 1990. Noun classification from predicate-argument structures. *Proc. of the ACL*, 268–275.

J. J. Jiang and D. W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *Proceedings of International Conference on Research on Computational Linguistics (ROCLING X)*, Taiwan.

Y. Jing and B. Croft. 1994. An Association Thesaurus for Information Retrieval. *Proc. of Recherche d'Informations Assistée par Ordinateur (RIAO)*, 146–160.

D. Lin. 1998. Automatic retrieval and clustering of similar words. *Proc. of COLING/ACL 1998*, 786–774.

L. Lee. 1999. Measures of distributional similarity. *Proc. of the ACL*, 23–32

L. Lee. 2001. On the Effectiveness of the Skew Divergence for Statistical Language Analysis. *Artificial Intelligence and Statistics 2001*, 65–72.

S. Mohammad and G. Hirst. 2006. Distributional measures of concept-distance: A task-oriented evaluation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Sydney, Australia.

P. Resnik. 1995. Using information content to evaluate semantic similarity. *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, 448–453, Montreal, Canada.

G. Ruge. 1997. Automatic detection of thesaurus relations for information retrieval applications. *Foundations of Computer Science: Potential - Theory - Cognition*, LNCS, Volume 1337, 499–506, Springer Verlag, Berlin, Germany.

S. Shalev-Shwartz, Y. Singer and A. Y. Ng. 2004. Online and Batch Learning of Pseudo-Metrics. *Proc. of the International Conference on Machine Learning (ICML)*.

M. Schutz and T. Joachims. 2003. Learning a Distance Metric from Relative Comparisons. *Advances in Neural Information Processing Systems (NIPS)*..

J. Weeds, D. Weir and D. McCarthy. 2004. Characterising Measures of Lexical Distributional Similarity. *Proc. of COLING 2004*, 1015–1021.

K. Q. Weinberger, J. Blitzer and L. K. Saul. 2005. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Advances in Neural Information Processing Systems (NIPS)*.

E. P. Xing, A. Y. Ng, M. Jordan and S. Russell 2002. Distance metric learning with application to clustering with sideinformation. *Advances in Neural Information Processing Systems (NIPS)*.

Y. Yang and J. O. Pedersen. 1997. A Comparative Study on Feature Selection in Text Categorization. *Proc. of the International Conference on Machine Learning (ICML)*, 412–420.