Vector based Approaches to Semantic Similarity Measures

Juan M. Huerta

IBM T. J. Watson Research Center
1101 Kitchawan Road, Yorktown Heights, NY, 10598
huerta@us.ibm.com

Abstract. This paper describes our approach to developing novel vector based measures of semantic similarity between a pair of sentences or utterances. Measures of this nature are useful not only in evaluating machine translation output, but also in other language understanding and information retrieval applications. We first describe the general family of existing vector based approaches to evaluating semantic similarity and their general properties. We illustrate how this family can be extended by means of discriminatively trained semantic feature weights. Finally, we explore the problem of rephrasing (i.e., addressing the question *is sentence X the rephrase of sentence Y?*) and present a new measure of the semantic linear equivalence between two sentences by means of a modified LSI approach based on the Generalized Singular Value Decomposition.

1 Introduction

Measurements of semantic similarity between a pair of sentences[1] provide a fundamental function in NLU, machine translation, information retrieval and voice based automation tasks, among many other applications. In machine translation, for example, one would like to quantitatively measure the quality of the translation output by measuring the effect that translation had in the conveyed message. In voice based automation tasks, for example in natural language call routing applications, one approach one could take is to compare the uttered input against a collection of canonical or template commands deeming the closest category as the intended target.

Current approaches to semantic similarity measurement include techniques that are specific or custom to the task at hand. For example, in machine translation, the BLEU metric [1] is used in measuring similarity of the MT output. In call routing, vector based methods (e.g., [2, 3]) are used to compare the input utterance against a set of template categories. In information retrieval some approaches use the cosine distance between a query and a document-vector mapped into a lower dimension LSI concept

[1] In this paper, for the sake of conciseness, we use the terms *document*, *utterance*, and *sentence* interchangeably. Typically the nature of the task define the specific type (for example, voice automation systems use *utterances* and so on).

Received 15/10/07
Accepted 07/12/07
Final Version 29/01/08

space ([4]) . Furthermore, IR-inspired methods are currently being applied to novel domains like question answering [9].

In this paper we introduce two novel vector based approaches to semantic similarity (namely, discriminatively trained semantic weights and a Generalized Singular Value Decomposition (GSVD) based approach) based on existing vector based approaches (specifically, LSI, cosine distance, BLEU and discriminative approaches). This paper is organized as follows, we first describe cosine distance, the BLEU metric and discriminative approaches and provide some background related to the need for weight inclusion. We then describe a novel approach to obtaining and applying discriminatively trained semantic weights on the features when computing these basic distances. Finally we introduce a novel method to measure semantic similarity based on a common concept space using the Generalized Singular Value Decomposition. We provide some illustrative examples and a set of preliminary experiments based on a rephrase corpus and on a chat corpus.

2 Vector Based Approaches

In this section we provide a brief overview of some existing vector based approaches to utterance similarity measurement or classification and provide some useful background to these metrics. The approaches described in this section are the building blocks for the novel techniques we introduce in sections 3 and 4.

Existing vector based approaches to document classification and retrieval can be categorized based on 3 criteria: (a) the feature type employed, (b) the weighting or functions applied on the feature counts, and (c) vector distance they use. For example, BLEU typically uses (a) n-gram features, (b) flat weight multipliers are applied at the class levels (i.e., one weight for unigram features, one for bigram features etc) and (c) the distance between two documents is an exponential function of modified precisions of the observed features. Another example is Vector Based call routing [3] which uses (a) n-gram features, (b) discriminatively trained weights in the classification matrix vectors, and normalized occurrence counts for the utterance vector and (c) cosine distance between topic matrix vectors and utterance vector.

2.1 Cosine Distance

The cosine distance is one of the simplest ways of computing similarity between two documents by measuring the normalized projection of one vector over the other. It is defined as follows,

$$\cos\theta = \frac{a \cdot b}{|a||b|}$$

One can compute the similarity between two documents simply by computing the cosine distance between their feature vectors. This approach, while coarse, is widely used in IR tasks and call routing tasks, for example. One of its main advantages is that it is domain and model free.

2.2 BLEU

Bleu [1] is a vector based metric intended to measure the quality of the machine translation output and has the important feature of correlating with human evaluator scores. BLEU is based on modified precision. It is defined below,

$$BLEU = \exp\left(\min(1-\frac{r}{c},0)+\sum_{n=1}^{N} w_n \log p_n \right)$$

$$= \exp(\min(1-r/c,0))p_4^{w_4} p_3^{w3} p_2^{w_2} p_1^{w_1}$$

Where c and r are the lengths of the candidate translation sentence and of the reference, respectively; and p_n denotes the modified precision, which is given by,

$$p_n = \frac{\sum\limits_{C\in\{Candidates\}} \sum\limits_{n-gram\in C} Counts_{clip}(n-gram)}{\sum\limits_{C\in\{Candidates\}} \sum\limits_{n-gram\in C} Counts\ (n-gram)}$$

BLEU is essentially the geometric average of weighted modified precisions multilplied by a non-linear term related to length penalty. One can observe that BLEU and cosine distance essentially could share the same features (i.e., vector based on the same n-grams) and could share the same weights. But how are modified precision related to cosine distance? We address this question in section 2.4 with the objective of gaining further insight.

2.3 Discriminant Approaches

MaxEnt (maximum entropy) approaches are designed to maximize the conditional likelihood [5] and thus belong in the discriminant approaches. The conditional likelihood is given here,

$$P(c_j \mid d,\lambda) = \frac{\exp\sum \lambda_{i,j} f_i(c_j,d)}{\sum_c \exp\sum \lambda_{i,j} f_i(c_j,d)}$$

While the classification rule is given below. Which can be explained as follows: if f is a feature vector the class J is the argument that maximizes the dot product of the Lambda matrix and f. In other words, classification in MaxEnt approaches can be seen as a dot product between lambda vectors and feature vectors and thus as a vector distance between observed features and template vectors.

$$\text{class J} = \arg\max_j \frac{\exp(\Lambda f)}{sum(\exp(\Lambda f))} = \arg\max_j \exp(\Lambda^j f) = \arg\max_j \lambda_i^j f_i$$

2.4 Relating Cosine and BLUE: Document Perturbation Perspective

So far, we have described the cosine distance and BLUE score, we additionally have explained that MaxEnt discriminative approaches to document classification can be expressed in terms of a cosine distance between query document and a classification matrix. Now we address the question of given a pair of documents, how much similarity there is between the cosine distance and the BLEU score. For this purpose we assume a document perturbation perspective: we assume a document a and a document b in which b is the *perturbed* version of a (i.e., the original document plus a small random variation). We then compute the cosine distance and the modified precision between a and b.

For the case of the cosine distance, we note that the feature counts a_i are always positive while the perturbation noise is a zero mean random variable, thus the cosine distance can be expressed as:

$a =$ document

$b = a + \varepsilon$ (perturbed document)

$$\cos\theta = \frac{a\cdot(a+\varepsilon)}{|a||a+\varepsilon|} = \frac{\sum a_i^2 + \sum a_i\varepsilon_i}{\left(\sum a_i^2\right)^{1/2}\left(\sum(a_i+\varepsilon_i)^2\right)^{1/2}}, \varepsilon \text{ is a zero mean Random Variable}$$

$$\cos\theta = \sqrt{\frac{\left(\sum a_i^2\right)}{\left(\sum a_i^2 + \sum \varepsilon_i^2\right)}}$$

For a large summation (i.e., large document) the sum of the square of alphas and of epsilons will become a Gaussian.

For the modified precisions, we express the clipped frequencies (i.e., frequency counts in the candidate that mach the reference, not exceeding the counts in the reference) as the original counts plus a random variable gamma which has with only positive values.

$$p = \frac{\sum a_i - \sum \gamma_i}{\sum a_i}, \gamma > 0 \text{ but } \sum \gamma_i \text{ is Gaussian for large sumation.}$$

From the equations above we can make two observations. The first observation is that the cosine distance and modified precision, which is the building block of BLEU, are most similar whe the difference between documents is relatively small or when the size of the document is very large (ratio of Gaussian Random Variables). The second observation is related to the independence assumption about the features a_i : the behavior of the perturbation, more clearly in the Cosine Distance case, has terms that average out to zero the more the a_i features are truly *Independent and Identically Distributed* (IID); in other words these two metrics blur out by averaging positive and negative perturbations. In reality, natural word frequencies tend to affect this assumption (i.e., not strictly IID), but naturally giving more weight to frequent words and this might not necessarily be a desirable characteristic since it will bias the metric towards frequent features. To minimize this, it is very important to emphasize meaningful words, if what we are interested in is in measuring similarity in the meaning of the utterances. In the next section we propose a method to address this.

3 Toward Discriminant Semantic Weights

In the previous chapter we have provided some background on Cosine Distance, BLEU and discriminative approaches. We illustrated the importance of introducing weights that emphasize semantic relevance and at the same time counterbalance the bias introduced by natural word frequencies. The question we address now is how to identify those words and how to train these weights and integrate them with the metric? We now describe a novel way to retrofit BLEU to incorporate discriminatively trained semantic weights.

In section 2.3 we described the criteria used in MaxEnt and noted that that if the set of weights used for a given feature across the set of classes has little *variance* (i.e., dynamic range), then the contribution of such feature to the overall classification is small. This is precisely what we will use to identify the semantic importance of a feature. In other words we focus on the dynamic range of such feature weight set,

$$w_{f_j} = \max(\lambda_{f_j,i}) - \max(\lambda_{f_j,i})$$

These weights tells us how much contribution to discrimination the feature provide and is always equal or larger than zero. The classification is done across semantic groupings, or classes thus these weights denote semantic-class discriminant importance. Thus we could adjust the BLEU metric to include these weights by making ,

$$\hat{p}_n = \frac{\sum\limits_{C\in\{Candidates\}} \sum\limits_{n-gram\in C} w_{n-gram} Counts_{clip}(n-gram)}{\sum\limits_{C\in\{Candidates\}} \sum\limits_{n-gram\in C} w_{n-gram} Counts\ (n-gram)}$$

In the case of cosine distance the weights are based on weighted versions of *a* and *b* and *W* is the diagonal matrix with the semantic weights on the diagonal.

$$Semantic\cos\theta = \frac{(W^{1/2}a)\cdot(W^{1/2}b)}{|W^{1/2}a||W^{1/2}b|}$$

Thus, our approach utilizes a labeled corpus in which semantic categories are indicated for each utterance. A discriminant criterion is then employed to obtain the matrix Λ from which the weight dynamic ranges are computed in order to obtain the semantic weights, which are used in BLEU and Cosine Distance to weight the feature perturbations accordingly.

While weighted versions of BLEU and the cosine distance have been previously proposed, (e.g., [6]) these approaches have not focused on discriminatively trained semantic weights.

4 Shared Concept Space Approach

We now propose a novel approach to measure utterance similarity in a pair of utterances when one utterance might be a rephrase of the other. We pose the problem as a classification problem: we assume the existence of a parallel corpus in which each sentence in set A has a rephrased version in the set B. While simple cosine distance between utterances in the set A and candidates in the set B can be used in this rephrase classification task, approaches that provide more robustness to noise and sparse input are desirable. The family of approaches to document similarity and document query similarity based on SVD in general and the Latent Semantic Analysis [4] in particular, are commonly used in information retrieval and document similarity evaluation (e.g., [7, 10]) and provide desirable features like the ones described. In LSI the document term matrix A is factored in the following way:

$$A = U\Sigma V^T$$

A query and a document can then be represented in a concept space (using k singular values) as shown below, the cosine distance in this concept space is used to compute the similarity between query and document or between documents.

$$\hat{q} = \Sigma_k^{-1} U_k^T q = \Lambda_k q$$

Our approach assumes, as we mentioned at the beginning of this section, the existence of two parallel document term matrices A and B. These two document-term matrices have the same number of columns (because they are parallel), but the number of rows (the lexicons each use) need not be the same. We could perform SVD on each matrix separately. However, we now describe a novel method based on the Generalized Singular Value decomposition [8] that attains decompositions of the matrices that share the same concept space.

4.1 Generalized Singular Value Decomposition

The Generalized Singular Value decomposition of matrices A and B, decomposes these matrices as follows,

$$A = UCX^T$$

$$B = VSX^T$$

Following an LSI interpretation, U and V represent the term to concept mapping matrices and X represents the document to concept matrix. Having matrix X shared by A and B achieves the desired tying of concept spaces.

The way to map documents or queries to concept space is then,

$$(C_k{}^T C_k)^{-1} C_k{}^T U^T A = X^T$$

$$(S_k{}^T S_k)^{-1} S_k{}^T V^T B = X^T$$

In the next section we present some illustrative examples and preliminary experiments of the concepts and techniques we have discussed so far.

5 Experiments

We describe now two sets of experiments we performed. The first one illustrates the discriminative semantic weights using a IT Help Desk corpus, while the second experiment is related to rephrase detection/classification using the GSVD approach.

5.1 Discriminant Weights Experiment

To illustrate the Discriminative Weight approach we used a corpus of text chats between IT users and Help Desk agents. The users and the agents communicate via instant messaging to troubleshoot the user's IT problems. The corpus consist of over 3000 chats. The semantic categories used in the discriminative training correspond the broad product category discussed (e.g., email, network connectivity, web applications, telephony, mainframe, etc.). We used Maximum Entropy training to obtain the classification matrix. Our features consisted uniquely on unigrams. Table 1 below show the outcome of the computation of the words with the highest dynamic range and the words with the lowest dynamic range. Thus, words that carry a substantial amount of semantic importance, in terms of message, are assigned high weights. In general these words describe products and items that can affect the classified semantic category substantially. On the other hand, the words with low dynamic range (and thus low weight in semantically weighted BLEU and cosine distance) are typically verbs which by themselves carry little discriminant semantic power and thus are less important, in this task, in terms of power to convey a message and affect a classification output.

Table 1. Words with highest dynamic range (left column) and lowest dynamic range (right column)

HIGH WEIGHT	LOW WEIGHT
GSA 2.860040	RULE 0.016409
MAIL 3.472591	KICKING 0.017700
BLUEPAGES 3.162921	SWITCHING 0.021317
MANAGENOW 2.502134	CONTAINED 0.013456
PRINTER 2.662830	WORTH 0.012854
EMAILS 3.210260	CHOOSING 0.013268
P/W 1.991220	MONITORING 0.010465
(LOTUS) NOTES 2.402410	
QUICKPLACE 2.626500	
DATABASE 2.775500	
TSM 2.148356	
AT&T 2.648001	

For illustration purposes, we now show a couple of synthetic examples in which BLEU and semantically weighted BLEU scores are compared depending on two sentences. The phrases used are made up and are not part of the corpus (nor representative of the corpus). Table 2 below shows the original sentence and the result of a translation and a roundtrip translation. The output of the back-translation is compared against the original and the scores are computed. We can see that when errors are introduced in important features (e.g., *wireless*) semantic bleu produces a lower score compared to BLEU. Conversely, when errors are introduced in non-important features (e.g., deletion) the score is higher than BLEU. Thus as intended, relative to BLEU, the semantic weighted BLEU produces a score that is more sensitive to perturbation if the perturbation is important, and less if the perturbation is unimportant.

Table 2. Sample phrases, original and perturbed versions, with their BLEU and semantically weighted BLEU scores for two cases.

BLEU > Sem. BLEU	BLEU < Sem. BLEU
Original: Please reset *wireless* connectivity **Perturbed:** That restores connectivity *without* *threads* please BLEU: 0.323 Sem. BLEU: 0.315	**Original:** Recent calendar deletion **Perturbed:** Recent calendar suppression BLEU: 0.757 Sem. BLEU: 0.792

5.2 Rephrasing Experiments

To conduct our rephrase experiments, we took 2 parallel versions of a portion of the Bible. Specifically we took the book of Proverbs. The structure of the verses in book of Proverbs (915 in total) is relatively simple. Because of the careful attention paid in keeping the message in every translation of the Bible, and because of the size and simplicity in the sentence structure of the verses of the Book of Proverbs, it presents an ideal corpus for preliminary experiment in rephrase analysis. The versions used are the New King James version and the New Life[2] version. The lexicon sizes are: 1956 words for the NKJ version, 1213 for NL, and 2319 for both version. The NKJ version has 14822 tokens while NL version has 17045 tokens. The NL version uses less unique words but it is longer, while NKJ has a bigger vocabulary while being shorter.

The first experiment that we conducted on this corpus is to measure self and cross utterance similarity across versions using the cosine distance. Figure 1 above shows the distribution of the cosine distance value between a sentence and all the other sentences in the other version (cross similarity, left panel) and on the right shows the self similarity which is the distribution of cosine scores between a sentence and its counterpart in the other version. As we can see, from the *detection* point of view, the task is relatively simple, as most of the cross similarity scores lie below 0.5 and most of the self similarity scores lie above 0.5. In terms of *classification* we performed the following experiment: we computed the cosine distance between each verse and the whole other version. If the highest cosine score belongs to the corresponding

[2] Scripture taken from the New King James Version. Copyright © 1982 by Thomas Nelson, Inc. Used by permission. All rights reserved

utterance in the other corpus we count one correct classification (and an incorrect otherwise). We did this both ways (classifying version A first and then classifying version B).The results are shown below in table 3. As we can see, about 80% accuracy is achieved with cosine distance.

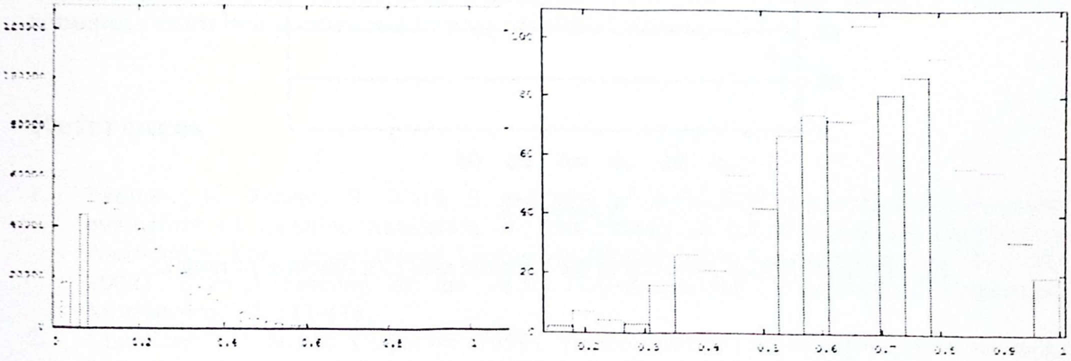

Fig. 1. Cross similarity (left) and self similarity (right) Cosine distance measure distribution.

Table 3. Classification results using the Cosine Distance

	Accuracy
A*b	80%
B*a	78%

We also performed GSVD experiments. We constructed the document-term matrices for each of the translation versions A and B. We then computed the generalized singular value decomposition described in section 4. Using that representation we mapped each sentence to a concept space using several rank values k. Then we repeated the cosine distance experiment described above with the sentences mapped to this concept space. The results are shown in figure 2 below. Interestingly an accuracy of about 99% is obtained with k as low as 26. And with k equal to 16 we get an accuracy comparable to plain cosine distance.

While these results are remarkable, a substantial accuracy improvement over cosine distance, one has to be careful to note that the SVD is performed on the whole corpus. Because of its very small size, it is impractical to perform a breakdown of the corpus into training and testing components and thus an experiment with a much larger corpus is needed. Furthermore, as we mentioned, the sentence constructions in the book of Proverbs are quite simple and thus the linear mappings captured in the GSVD are good enough to attain such high accuracy.

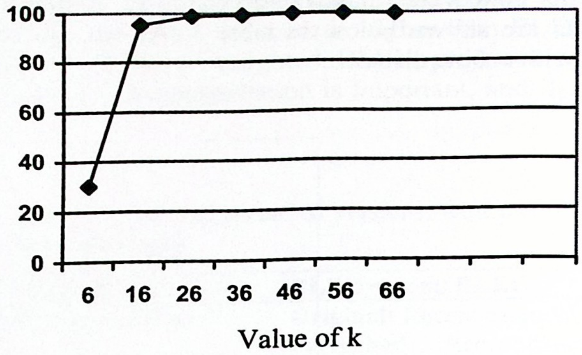

Fig. 2. Classification accuracy in the rephrase task as a function of rank *k*.

6 Discussion

In this paper we reviewed several important vector based approaches to computing similarity between two sentences, documents or utterances: cosine distance, BLEU, discriminative vector based methods, and SVD based methods (like LSI). Besides describing their basic characteristics, in this paper, we pointed out that semantic weighting is important to avoid the undesirable averaging out effect observed in perturbation analysis. We proposed a method to discriminatively train these weights and described ways to incorporate these weights into cosine distance and BLEU. Through some illustrative examples we saw that when contrasting semantically weighted BLEU with BLEU we can provide an improved guidance the semantic distance.

We also analyzed SVD approaches and proposed a novel utilization of the generalized singular value decomposition to vector based computation of semantic similarity. We concluded the paper with a series of illustrative examples and preliminary experiments.

We observed that in our preliminary experiments the basic cosine distance had a classification accuracy of 80% in the rephrase task, while the GSVD based approach performed at around 99%. This result, while very interesting, might be due mostly to linear mappings between rephrases (i.e., the use of features that can be substituted by other features, like synonyms) which might be due to the nature of the simple structure formations and almost deterministic mappings of the sentences conforming the corpus. We pointed out in our experiment section that, in the case of rephrase analysis, it is important to conduct further experiments on corpora that presents larger sentence construction variability and that is large enough to cover a larger lexicon. While the preliminary results we obtained seemed very promising, it will be of much better value to test this approach on a very large set.

In terms of future work and future approaches, we suggest the exploration of directions that are probabilistic in nature. In this paper we proposed extensions to BLEU, SVD-LSI, and cosine distance. These approaches are not probabilistic in nature. Dis criminative methods, and Exponential Models-Maximum Entropy approaches are the only type of probabilistic approaches used in this paper. The authors consider that the methods introduced in this paper would benefit in terms of robustess from being extended into a probabilistic formulation.

References

1. Papineni, K., Roukos, S., Ward, T., and Zhu, W. 2001. BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting on Association For Computational Linguistics (Philadelphia, Pennsylvania, July 07 - 12, 2002). Annual Meeting of the ACL. Association for Computational Linguistics, Morristown, NJ, 311-318.
2. Chu-Carroll, J, and R. Carpenter (1999), Vector-Based Natural Language Call Routing. Journal of Computational Linguistics, 25(30), pp. 361-388, 1999
3. H.-K.J. Kuo,; Chin-Hui Lee Discriminative training of natural language call routers Speech and Audio Processing, IEEE Transactions on Volume 11, Issue 1, Jan 2003 Page(s): 24 - 35.
4. Thomas Landauer, P. W. Foltz, & D. Laham (1998). "Introduction to Latent Semantic Analysis". Discourse Processes 25: 259-284.
5. Berger, A. L., Pietra, V. J., and Pietra, S. A. 1996. A maximum entropy approach to natural language processing. Comput. Linguist. 22, 1 (Mar. 1996), 39-71.
6. J.M. Schultz and M. Liberman, "Topic Detection and Tracking using idfWeighted Cosine Coefficient," Proceedings of the DARPA Broadcast News Workshop, 189-192, 1999.
7. Dasgupta, A., Kumar, R., Raghavan, P., and Tomkins, A. 2005. Variable latent semantic indexing. In Proceeding of the Eleventh ACM SIGKDD international Conference on Knowledge Discovery in Data Mining (Chicago, Illinois, USA, August 21 - 24, 2005). KDD '05. ACM Press, New York, NY, 13-21.
8. Gene Golub, and Charles Van Loan, Matrix Computations, Third Edition, Johns Hopkins University Press, Baltimore, 1996,
9. Gregory Marton and Boris Katz Using Semantic Overlap Scoring in Answering TREC Relationship Questions, Proceedings of LREC 2006, Genoa, Italy; May, 2006
10. Evgeniy Gabrilovich and Shaul Markovitch Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis Proceedings of The 20th International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India, January 2007