Title
NLS: A Non-Latent Similarity Algorithm

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 26(26)

ISSN
1069-7977

Authors
Cai, Zhinqiang
McNamara, Danielle S.
Louwerse, Max
et al.

Publication Date
2004

Peer reviewed

NLS: A Non-Latent Similarity Algorithm

Zhiqiang Cai (zcai@memphis.edu)

Danielle S. McNamara (dsmcnamr@memphis.edu)

Max Louwerse (mlouwers@memphis.edu)

Xiangen Hu (xhu@memphis.edu)

Mike Rowe (mprowe@memphis.edu)

Arthur C. Graesser (a-graesser@memphis.edu)

Department of Psychology/Institute for Intelligent Systems, 365 Innovation Drive
Memphis, TN 38152 USA

Abstract

This paper introduces a new algorithm for calculating semantic similarity within and between texts. We refer to this algorithm as NLS, for Non-Latent Similarity. This algorithm makes use of a second-order similarity matrix (SOM) based on the cosine of the vectors from a first-order (non-latent) matrix. This first-order matrix (FOM) could be generated in any number of ways; here we used a method modified from Lin (1998). Our question regarded the ability of NLS to predict word associations. We compared NLS to both Latent Semantic Analysis (LSA) and the FOM. Across two sets of norms, we found that LSA, NLS, and FOM were equally predictive of associates to modifiers and verbs. However, the NLS and FOM algorithms better predicted associates to nouns than did LSA.

Introduction

Computationally determining the semantic similarity between textual units (words, sentences, chapters, etc.) has become essential in a variety of applications, including web searches and question answering systems. One specific example is AutoTutor, an intelligent tutoring system in which the meaning of a student answer is compared with the meaning of an expert answer (Graesser, P. Wiemer-Hastings, K. Wiemer-Hastings, Harter, Person, & the TRG, 2000). In another application, called Coh-Metrix, semantic similarity is used to calculate the cohesion in text by determining the extent of overlap between sentences and paragraphs (Graesser, McNamara, Louwerse & Cai, in press; McNamara, Louwerse, & Graesser, 2002).

Semantic similarity measures can be classified into Boolean systems, vector space models, and probabilistic models (Baeza-Yates & Ribeiro-Neto, 1999; Manning & Schütze, 2002). This paper focuses on vector space models. Our specific goal is to compare Latent Semantic Analysis (LSA, Landauer & Dumais, 1997) to an alternative algorithm called Non-Latent Similarity (NLS). This NLS algorithm makes use of a second-order similarity matrix (SOM). Essentially, a SOM is created using the cosine of the vectors from a first-order (non-latent) matrix. This first-order matrix (FOM) could be generated in any number of ways. However, here we used a method modified from Lin (1998). In the following sections, we describe the general concept behind vector space models, describe the differences between the metrics examined, and present an evaluation of these metrics' ability to predict word associates.

Vector Space Models

The basic assumption behind vector space models is that words that share similar contexts will have similar vector representations. Since texts consist of words, similar words will form similar texts. Therefore, the meaning of a text is represented by the sum of the vectors corresponding to the words that form the text. Furthermore, the similarity of two texts can be measured by the cosine of the angle between two vectors representing the two texts (see Figure 1).

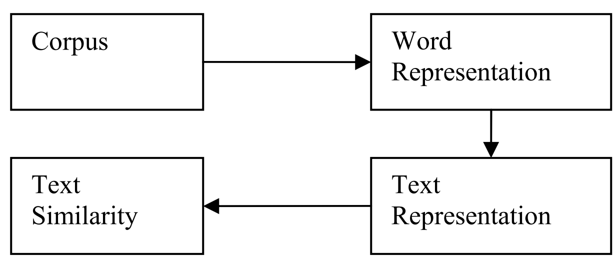

Figure 1. From Corpus to Text Similarity.

The four items of Figure 1 can be described as follows. First, the corpus is the collection of words comprising the target texts. Second, word representation is a matrix G used to represent all words. Each word is represented by a row vector g of the matrix G. Each column of G is considered a "feature". However, it is not always clear what these features are. Third, text representation is the vector $v = G^{T}a$ representing a given text, where each entry of a is the number of occurrences of the corresponding word in the text. Fourth, text similarity is represented by a cosine value between two vectors.

More specifically, Equation 1 can be used to measure the similarity between two texts represented by a and b, respectively. For reasons of clarity, we do not include word weighting in this formula.

$$sim(a,b) = \frac{a^T GG^T b}{\sqrt{a^T GG^T a}\sqrt{b^T GG^T b}} \quad (1)$$

Latent Semantic Analysis (LSA)

LSA is one type of vector-space model that is used to represent world knowledge (Landauer & Dumais, 1997). LSA extracts quantitative information about the co-occurrences of words in documents (paragraphs and sentences) and translates this into an *N*-dimensional space. The input of LSA is a large co-occurrence matrix that specifies the frequency of each word in a document. Using singular value decomposition (SVD), LSA maps each document and word into a lower dimensional space. In this way, the extremely large co-occurrence matrix is typically reduced to about 300 dimensions. Each word then becomes a weighted vector on *K* dimensions. The semantic relationship between words can be estimated by taking the cosine between two vectors. This algorithm can be briefly described as follows.

(1) Find the word-document occurrence matrix *A* from a corpus[1].
(2) Apply SVD: $A = U\Sigma V^T$.
(3) Take the row vectors of the matrix *U* as the vector representations of words.

Non-Latent Similarity (NLS) Model

NLS is proposed here as an alternative to latent similarity models such as LSA. NLS relies on a first order, non-latent matrix that represents the non-latent associations between words. The similarity between words (and documents) is calculated based on a second-order matrix. The second order matrix is created from the cosines between the vectors for each word drawn from the FOM. Hence, for NLS, the cosines are calculated based on the non-latent similarities between the words, whereas for LSA, the similarities are based on the cosines between the latent vector representations of the words. The following section describes the components and algorithms used in NLS.

Lin's (1998) Algorithm Our starting point for NLS is Lin's (1998) algorithm for extracting the similarity of words. Similarity is based upon the syntactic roles words play in the corpus. A syntactic role is designated here as a feature. For example, "the Modifier of the NP *man*" is a feature. A word has this feature if and only if it is used as the modifier of *man* when *man* is part of an NP in the corpus. For example, if the corpus contains the phrase *the rich man*, then *rich* has the (adjectival) feature of modifying *man.* Each feature is assigned a weight to indicate the feature's importance. This algorithm is briefly described as follows.

(1) For each word base, form a feature vector.
(2) For each pair of word bases, find the similarity of two word bases from the corresponding two feature vectors.

In Lin's algorithm, the similarity is calculated according to Equation 2.

$$sim(w_1, w_2) = \frac{2 \times I(F(w_1) \cap F(w_2))}{I(F(w_1)) + I(F(w_2))} \quad (2)$$

$F(w)$ is the set of features possessed by the word w and $I(F)$ is the "information" contained in the feature set F: $I(F) = \sum_{f \in F} u(f)$. u is the weight function of the feature f.

First-Order Matrix LSA is referred to as latent because the content is not explicit or extractable after SVD. Thus, the features that two similar words share are "latent." In contrast, every feature is explicit and directly extractable from the matrix using Lin's (1998) algorithm. Hence, it is non-latent, and can be used as a first-order similarity matrix (FOM).

We created the FOM using a modification of Lin's algorithm with cosines rather than proportions. First, we parsed all of the sentences (about 2 million) in the TASA corpus using Lin's MINIPAR parser (Lin, 1998). This provided about 9 million word-relation-word triplets. Table 1 shows the triplets extracted for the sentence *People did live in Asia, however*.

Table 1: An example with word-relation-word triplets.

Word1	Relation	Word2
Live	V:s:N	people
Live	V:aux:Aux	do
Live	V:mod:Prep	in
In	Prep:pcomp-n:N	Asia
Live	V:mod:A	however

A "feature" consists of a word (e.g., Word1 or Word2) and a relation that contains a verb (V), noun (N), or modifier (A). For example, the association between the word *live* and its relation to *people*, which is "V:s:N", comprises two features (*live* - V:s:N; *people* - V:s:N). About 400,000 such features were obtained. Each feature was assigned a weight, using Lin's formula. We adopted an occurrence frequency threshold, which yielded 10363 nouns (occurrence > 50), 5687 verbs (occurrence > 5), and 6890 modifiers (occurrence > 10). For each of the selected words, a feature vector was formed according to the features it involved.

We modified Lin's method in the last step. Specifically, rather than applying Equation 2 to the feature vectors, the cosine between any two feature vectors was calculated. This provided a FOM containing the similarity between all word pairs. In addition, the FOM guarantees a property called "decomposability", which will be addressed in the next section.

[1] Hu et al. (2003, theorem 2) proved that the LSA similarity measure is a special case of (1)

Non-Latent Similarity (NLS) Algorithm The logic behind the use of a second order matrix to represent textual similarity relies on a reformulation of the algorithm used in general vector models. Specifically, Equation 1 can be rewritten as Equation 3.

$$sim(a,b) = \frac{a^T Sb}{\sqrt{a^T Sa}\sqrt{b^T Sb}} \qquad (3)$$

When the columns of G are normalized to be unit vectors, S becomes a word-similarity matrix [2]. In other words, each entry of $S, s_{ij} = g_i g_j^T$, is the similarity of two words represented by row vectors g_i and g_j, respectively. Essentially, a word-similarity matrix (S) is used rather than word representation vectors (G).

From Equation 3 we can see that the similarity of two texts is determined by two factors: the word occurrences in each text and the similarity between words. Since we can do little to the occurrence vectors (other than applying word weighting), the word similarity matrix will determine the validity of the measure of text similarity. In other words, Equation 3 provides a good measure if and only if similar words have similar vector representations. If similar words have dissimilar vector representations or dissimilar words have similar representations, then the measure provided by Equation 3 is unreliable. Therefore, the verification of the validity of the word representation, at least in terms of text similarity comparison, is equivalent to the verification of the validity of the word similarity matrix (or FOM in this case).

While it is not possible to directly judge the quality of a vector representation, it is possible to judge the validity of word similarity. Provisions for such a judgment will be made in the next section of this paper.

Equation 3 raises an important question: Instead of creating the similarity matrix S by the word representation matrix G, can we find the similarity matrix by any other method that provides a better word similarity measure? One of the conditions under which this question may be answered is that the similarity matrix S, no matter how it is created, must be decomposable. That is, there exists a matrix G (we do not have to find it) such that $S = GG^T$. This condition is necessary to guarantee that the value calculated from Equation 3 ranges from -1 to 1.

The FOM that we generated by the modified Lin's method is decomposable and can therefore be used in Equation 3 for text comparison. However, that matrix is high-dimensional (N by N, where N is the total number of words). This will cause some computational complexity. To reduce the number of dimensions, we kept only the 400 largest similarity values for each word and set the other smaller values to be zero. Thus, the similarity matrix became sparse and the computational complexity was reduced. However, this made the similarity matrix un-decomposable and invalid for Equation 3.

[2] The normalization guarantees that the similarity between any two words will not exceed the similarity of a word to itself and that the values are in a known range [-1,1].

The decomposability therefore raises a new question: Is there a straightforward way to guarantee both decomposability and validity of the similarity matrix S? An easy way of guaranteeing these criteria is by using a word similarity matrix to act as a word representation matrix.

Suppose S is a word similarity matrix regardless of its creation method. Then each column vector in S contains the similarities of a particular word to all other words. Therefore, each column vector can represent the corresponding word.

Table 2. A small section of a first order matrix.

	chair	table	strength
desk	0.16	0.17	0
bed	0.14	0.13	0
speed	0	0	0.14
success	0	0	0.11

Table 2 is a small section of our FOM. It can be seen that the column vectors for *chair* and *table* are very similar to each other, but quite different from that of "strength". In the complete matrix, *desk* is the 4th nearest neighbor of (i.e., most similar to) *chair* and the 1st nearest neighbor of *table*. In addition, *bed* is the 2nd nearest neighbor of *chair* and the 5th nearest neighbor of *table* (see http://cohmetrix.memphis.edu/wordsim/wf1.aspx).

If we believe that similar words should share most nearest neighbors (a group of words that are most similar to a given word), then similar words should have similar column vectors in S. Therefore, we can create a new word similarity matrix by the cosine between the column vectors of S, $\tilde{S} = D^T S^T SD$, where D is a diagonal matrix formed by the reciprocal of the norms of the column vectors of S. We call $\tilde{S}$ the second-order word similarity matrix (SOM) and S the first-order similarity matrix (FOM). This new matrix $\tilde{S}$ is obviously decomposable and should maintain the validity of the original word similarity matrix.

If the SOM is valid, then we can form a measure based on the FOM:

$$sim(a,b) = \frac{a^T D^T S^T SDb}{\sqrt{a^T D^T S^T SDa}\sqrt{b^T D^T S^T SDb}} \qquad (4)$$

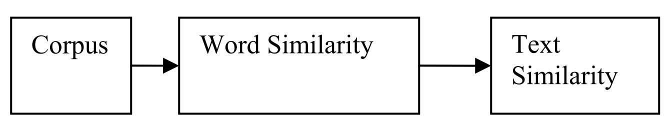

Figure 2. From Corpus to Text Similarity (SOM).

Equation 4 provides a new algorithm for text comparison, which relies solely on the similarity matrix. We call this algorithm the Non-Latent Similarity (NLS) algorithm, assuming that the FOM is non-latent. Figure 2 shows the difference between NLS and the general vector-space model. When compared with Figure 1, we can see that the "representations" are replaced by the similarity matrix.

Evaluation

In this section, we compare NLS to LSA to examine the differences between the latent analytic method exemplified by LSA and the non-latent method of NLS. We examine the validity of these two methods by examining their ability to predict word associates obtained from two sources of free association norms. We also examine the ability of the FOM to predict these word associates. The ability of FOM and NLS to predict word associates should be reflective of the overall validity of NLS to predict similarity of text corpora, which is crucial to our new algorithm shown in Equation 4.

We have two concerns. First, is our FOM valid? Second, if our FOM is valid, then will the second order similarity matrix (SOM) be valid as well? To answer these questions, we compared the validity of the following three similarity matrices generated by three different methods.

- *LSA*: The similarity matrix created from TASA corpus by LSA.
- *FOM*: The similarity matrix created from TASA corpus using the modified version of Lin's method.
- *NLS*: The second order similarity matrix based on the above *FOM*.

Our overall question addressed the ability of the three similarity metrics (LSA, FOM, and NLS) to correctly list word associates. We were also interested in examining how this ability varied as a function of several variables. First, we were interested in whether the results remained stable across norming databases. We chose to use two sets of free association norms: the Edinburgh Associative Thesaurus (EAT; Kiss, Armstrong, Milroy, & Piper, 1973) and the University of South Florida Free Association Norms (USFFAN; Nelson, McEvoy, & Schreiber, 1998).

We were also interested in how the results differed across word types (i.e., nouns, verbs, vs. adjectival/adverbial modifiers). One difference between the three classes of words is the amount of semantic contextualization. Specifically, the meaning of verbs and modifiers is usually context dependent, whereas the meaning of nouns is less dependent on the context (e. g., Graesser, Hopkinson & Schmid, 1987). For example, in the phrase *a big house*, the size of the adjectival modifier *big* depends on the noun *house*. It could be argued, moreover, that words that are more concrete are less context-dependent. Adjectives are less concrete than nouns so they would be more context-dependent. A similar argument could be made for verbs, which are more context dependent than nouns.

We expected the context-dependency factor to most affect the performance of LSA, because the success of LSA relies heavily on the occurrence of words in similar contexts, and essentially taps into that factor to assess word similarity. The basic assumption behind LSA is that words used in similar context have similar representations. Thus, if a word is less context-dependent, LSA may be less able to tap into associations.

While NLS similarly uses semantic context to compute similarity, it also uses syntactic context. The word similarities are extracted not only from the similar semantic context but also from the similar syntactic roles that the words play. That is, the FOM includes syntactic relations as features, whereas word order and the relations between words are ignored in LSA. Thus, we expected LSA to be less successful in identifying the associates of nouns as compared to modifiers and verbs. We did not expect this factor to affect the performance of NLS. We expected that FOM and NLS would be sensitive to both context based and non-context based associations.

To examine these factors, we randomly selected 135 common words, composed of 45 modifiers (including adjectives and adverbs), 45 nouns, and 45 verbs. We then determined the first most commonly listed and the second most commonly listed associate to those words, based on the association norms provided by EAT and the USFFAN. Finally, we determined whether each of the three similarity metrics listed the first and second most commonly listed associate from the respective norm database. A criterion was set in the following analyses: A metric identifies an associate of a word if, according to the metric, the associate is among the top five nearest neighbors of the word. While not extremely strict, the cutoff was intended to be relatively conservative compared to setting the cutoff at 20 words.

Results

Table 3 shows the proportion associates identified by each metric. A 3 x 2 x 2 analysis of variance (ANOVA) was conducted that included the between-words variable of word type (noun, verb, adjectival/adverbial modifier) and the within-words variables of associate (first, second) and database (EAT, USFFAN).

Table 3: Proportion of correctly identified associates listed in the top five nearest neighbors provided by LSA, FOM, and NLS as a function of the free association norms and word types.

	EAT			USFFAN		
	Mod	Noun	Verb	Mod	Noun	Verb
Associate 1						
LSA	0.40	0.11	0.16	0.31	0.07	0.13
FOM	0.40	0.36	0.13	0.31	0.31	0.16
NLS	0.38	0.36	0.11	0.27	0.36	0.13
Associate 2						
LSA	0.07	0.04	0.09	0.13	0.04	0.18
FOM	0.18	0.11	0.11	0.16	0.16	0.11
NLS	0.16	0.11	0.11	0.13	0.13	0.16

There was a main effect of word type, $F(2, 132) = 3.4$, $MSE = .471$, $p < .05$. Bonferroni Means tests indicated that the proportion of associates identified for modifiers ($M = .243$) was significantly greater than for verbs ($M = .122$), but not significantly greater than for nouns ($M = .187$). There was an effect of associate, $F(1, 131) = 19.5$, $MSE = .330$, $p<.001$, reflecting a greater proportion of first associates identified (M = .250) than second associates (M = 0.120). There was also an interaction between word type and associate, $F(2, 131) = 4.2$, $p < .05$. This interaction reflected an effect of word type for first associates, $F(2,132) = 5.5$, $MSE = .533$, $p<.01$ ($M_{modifier} = .34$ $M_{noun} = .26$ $M_{verb} = .14$), compared to no differences between word types for second associates, $F<1$, ($M_{modifier} = .14$, $M_{noun} = .11$, $M_{verb} = .12$). Thus, the metrics were unable to identify the second associates, regardless of word type.

Finally, there was significant effect of similarity metric, $F(2,264) = 4.6$, $MSE = .139$, $p <.05$, and an interaction between metric and word type, $F(4,264) = 4.1$, $p<.01$. This interaction is depicted in Figure 3. The interaction reflects the finding that the three metrics were equally successful in identifying the associates to modifiers and verbs, whereas FOM and NLS were significantly more successful in identifying the associates to nouns than was LSA, $F(2,88)= 4.1$, $MSE = .052$, $p < .05$.

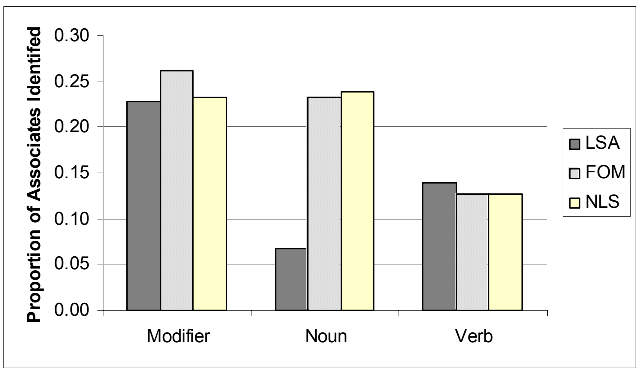

Figure 3. Proportion of associates identified (in the top 5 of the list) by the three similarity metrics.

These results did not depend on where the cutoff was drawn, (e.g., top 5 vs. top 20). Of course, the means increased with a more lax cutoff. For example, the overall accuracy of associate identification for LSA increased from 20% to 28% when the cutoff was set at 20 (i.e., when 20 of the words output by LSA were considered). Similarly, the overall accuracy for NLS increased from 27% to 42% when the cutoff was set at 20 words. Thus, there was a 140% and 157% increase respectively for LSA and NLS. The results also remained the same when word frequency was entered as a covariate. Essentially, these trends emerged regardless of how we examined the data.

There were no differences as a function of norming database. This indicates that the results we have reported should remain stable across norming databases.

Conclusions

In summary, we have provided an alternative algorithm, NLS, which makes it possible to use any non-latent similarity matrix to compare text similarity. This algorithm uses a second-order similarity matrix (SOM) that is created using the cosine of the vectors from a first-order (non-latent) matrix. This FOM could be generated in any number of ways. We used a modified form of Lin's (1998) algorithm to extract non-latent word similarity from corpora. Our evaluation of NLS compared its ability to predict word associates to the predictions made by the FOM and LSA. The critical difference between the algorithms addressed the latency of the word representations. The use of SVD results in latent word representations in LSA, whereas the use of the syntax parser in NLS results in a non-latent representation. We found that NLS, using the similarity matrix that we generated, identified the associates to modifiers and nouns relatively well. Both LSA and NLS were equally able to identify the associations to the modifiers. In contrast, none of the metrics successfully identified the associates to the verbs.

FOM versus NLS

There were two motivations for examining the results from the FOM as well as NLS. The first was to examine the validity of using a FOM. The second was to examine the correspondence in results between FOM and NLS. That is, if the FOM is valid, is the SOM valid as well? We found that NLS and FOM were equally successful in identifying all types of associations. This result indicates that SOM maintains the validity of FOM. The result supports the validity of using the NLS algorithm.

One consideration is that the second order similarity matrix may reveal new similarity relations which do not exist or are weak in the FOM. It is not hard to imagine that two words that have weak similarity in FOM may share some nearest neighbors and thus reveal a stronger relation between the two words in SOM. Nonetheless, we found here that the second-order matrix maintains the validity of FOM as much as possible, assuming the FOM is valid. When the FOM is decomposable, it can be directly used in NLS. The SOM is used when FOM is computationally heavy or is not decomposable. Our future investigations will work toward a better understanding of the situations that require a SOM as opposed to a FOM, or vice versa.

LSA versus NLS and FOM

We confirmed our predicted results that LSA would be less accurate in identifying the associates to context-independent nouns than to adjectival or adverbial modifiers, which have greater context dependency. We further predicted that this difference would not occur for NLS and the FOM. Indeed, NLS and FOM were equally predictive of noun and modifier associates. Thus, one advantage of NLS is that it makes use of both semantic and syntactic information within the text corpora. Specifically, the FOM includes both syntactic and semantic relations as features. Here, we have documented this advantage solely with respect to word similarities. However, we expect that this advantage will also improve the detection of similarity across larger bodies of text.

Verbs versus Nouns and Modifiers

One result that has baffled us is why NLS and LSA are both unable to pick up on the associates to the verbs. We considered several explanations. First, one might think that the number of forms of the word would be a factor to consider. Since verbs tend to have more forms than do modifiers (e.g., *add* has four forms: *add, added, adding, adds*), a typical vector space model would contain relatively less information about any one form of the verb. This factor may explain the inability of LSA to identify the associates to verbs. However, it cannot do so for NLS because we used the word base, not the word itself, when forming the matrix.

We further considered that humans may have produced a greater variety of associates to verbs than to nouns or modifiers. If so, then across the two databases (i.e., EAT and USFFAN), the match between the associates in one database to another should vary as a function of word type. However, this was not the case. The two databases matched the first associate for 69% of the words, with no differences across word types. There was lower agreement (40%) and greater variance for the second associate, but not in the expected direction.

An alternative explanation regards the contextualization of verbs as compared to nouns. As we stated earlier, the meaning of verbs is more dependent on semantic context than are nouns. In addition, verbs seem to be used in a wider variety of contexts. Whereas a person can do only so much with a *chair*, the person can *sit* just about anywhere and anyhow. One can imagine eating, walking, and thinking in any number of contexts, whereas the contexts for chairs and cars are more constrained. Hence, semantic context is more variable for verbs than for nouns. This variability may render models such as NLS or LSA unable to determine the 'meaning' of verbs.

This idea is in line with notions of how verbs are represented with semantic representations. Generally, verbs are treated as the links between the concepts. Verbs constitute the relations or links between nodes. Essentially, we see here that vector space models are less able to abstract meanings of relations than the meanings of concepts.

This notion gains clarity when we examine the associates to verbs that were provided by LSA and NLS. The EAT associates to *try* are *attempt* and *again*. LSA's top five predictions were *do*, *if, you*, *can*, and *wa*y. FOM's predictions were *think*, *say*, *go*, *know*, and *ask*. We can provide many examples such as these where the associates produced by the metric make little sense. The associations predicted for nouns and modifiers, in contrast, showed obvious relationships to the target word. This observation leads us to conclude that these metrics are not able to use contextual information of verbs, perhaps because that information is not available.

Acknowledgments

The research was supported by grants from DoD Multidisciplinary University Research Initiative (MURI) program administered by the Office of Naval Research (N00014-00-1-0600), National Science Foundation (SBR 9720314, REC0106965, ITR0325428), and the Institute of Education Sciences (IES R3056020018-02). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DOD, NSF or IES. The TASA corpus used was generously provided by Touchtone Applied Science Associates, Newburg, NY, who developed it for data on which to base their Educators Word Frequency Guide.

References

Baeza-Yates, R., Ribeiro-Neto, B. (Eds.) (1999). *Modern Information Retrieval*. New York , ACM Press.

Graesser, A. C., Hopkinson, P. L. & Schmid, C. (1987). Differences in interconcept organization between nouns and verbs. *Journal of Memory and Language*, *26*, 242-253.

Graesser, A.C., McNamara, D.S., Louwerse, M.M., & Cai, Z. (in press). Coh-Metrix: Analysis of text on cohesion and language. *Behavioral Research Methods, Instruments, and Computers.*

Graesser, A.C., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N., & the TRG (2000). Using latent semantic analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments, 8,* 129-148.

Hu, X., Cai, Z., Franceschetti, D., Penumatsa, P., Graesser, A.C., Louwerse, M.M., McNamara, D.S., & TRG (2003). LSA: The first dimension and dimensional weighting. *Proceedings of the 25th Annual Conference of the Cognitive Science Society* (pp. 587-592). Boston, MA: Cognitive Science Society.

Kiss, G.R., Armstrong, C., Milroy, R. & Piper, J. (1973). An associative thesaurus of English and its computer analysis. In A.J. Aitkin, R.W. Bailey & N. Hamilton-Smith (Eds.), *The computer and literary studies***.** Edinburgh : University Press.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The Latent Semantic Analysis. *Psychological Review*, *104*, 211-240.

Lin, D. (1998). An information-theoretic definition of similarity. *Proceedings of International Conference on Machine Learning* (pp. 296-304), Madison, Wisconsin.

Manning, C.D. & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: MIT press.

McNamara, D.S., Louwerse, M.M. & Graesser, A.C. (2002). *Coh-Metrix: Automated cohesion and coherence scores to predict text readability and facilitate comprehension*. Technical report, Institute for Intelligent Systems, University of Memphis, Memphis, TN.

Nelson, D. L., McEvoy, C. L., & Schreiber, T. A. (1998). *The University of South Florida word association, rhyme, and word fragment norms*. http://www.usf.edu/FreeAssociation/.