Combining Word Representations for Measuring Word Relatedness and Similarity

Nobal B. Niraula, Dipesh Gautam,
Rajendra Banjade, Nabin Maharjan, Vasile Rus
Department of Computer Science
Institute for Intelligent Systems
The University of Memphis
{nbnraula, dgautam, rbanjade, nmharjan, vrus}@memphis.edu

Abstract

Many unsupervised methods, such as Latent Semantic Analysis and Latent Dirichlet Allocation, have been proposed to automatically infer word representations in the form of a vector. By representing a word by a vector, one can exploit the power of vector algebra to solve many Natural Language Processing tasks *e.g.* by computing the cosine similarity between the corresponding word vectors the semantic similarity between the two words can be captured. In this paper, we hypothesize that combining different word representations complements the coverage of semantic aspects of a word and thus better represents the word than the individual representations. To this end, we present two approaches of combining word representations obtained from many heterogeneous sources. We also report empirical results for word-to-word semantic similarity and relatedness by using the new representation using two existing benchmark datasets.

Introduction

The task of measuring semantic similarity between two texts quantifies the degree of meaning they share together. Measuring such similarity score between two texts has tremendous usage in many Natural Language Processing (NLP) tasks. For instance, in paraphrase detection, finding a similar text corresponding to a given text requires computing the semantic similarity between the texts (Socher et al. 2011). In Intelligent Tutoring Systems, assessing students' answers relies on the similarity scores between their answers and the ideal answer provided by an expert (Rus and Graesser 2006). In information retrieval, finding semantically similar documents corresponding to a given query and finding similar queries requires semantic similarity between texts (Hliaoutakis et al. 2006). Many other tasks rely on computing semantic similarity between semantic similarity between two texts such as plagiarism detection(Osman et al. 2012), near duplicate document detection (Bayardo, Ma, and Srikant 2007), textual entailment (Dagan, Glickman, and Magnini 2006).

Despite having wide applications, computing semantic similarity between words/texts has been a long standing problem in NLP. Basically, two types of measures are used: *similarity* and *relatedness* measures. Although they are related, there are subtle differences between them. For instance, *chicken* and *egg* are related as they often appear together, but they are not similar (living vs non-living). Thus, similarity focused measures quantify the meaning shared by two words and relatedness focused methods quantify the associations between the words. In this paper, we present a model that will be evaluated for both types of measures.

There are two typical approaches people apply to compute the similarity between texts. The first approach computes the similarity directly. For instance, by representing a text by a vector, one can compute the semantic similarity between two texts by obtaining the cosine similarity between their vectors. The second approach relies on word-to-word similarity scores to compute text-to-text similarity at various levels. The idea is that a text consists of words and computing the semantic similarity score between two texts can be modeled by combining the semantic similarity scores between word pairs formed using the texts. Once the similarity scores between word pairs are obtained, a number of composition methods can be used to get the similarity between the texts (Rus and Lintean 2012; Niraula et al. 2013). In this regard, word-to-word similarity measure is the foundation of computing text-to-text similarity. Consequently, the research community is constantly exploring, such as our effort in this paper, to find better methods for word-to-word similarity. The stronger the correlation with human judgments, the better a method is.

Many unsupervised approaches such as Latent Semantic Analysis (LSA) (Landauer et al. 2007), Latent Dirichlet Analysis (LDA) (Blei, Ng, and Jordan 2003), and Vector Space Model (VSM) have been proposed to capture the meaning of words from a large collection of text corpora. For example, LSA can compute a semantic space of a chosen dimension, say M. With that, a word is represented by a M-dimensional vector. In LDA, a document is a distribution over topics and each topic is a distribution over words. It means a word in LDA can be represented by a vector with the number of topics as dimensions and the contribution of the word to the topics as the weights.

A resource that statistical machine learning models often use to learn word representations (i.e. vectors) is the Wikipedia (http://en.wikipedia.org), a huge collection of text articles. Once words are represented by vectors, the power of vector algebra can be exploited. For instance, a text containing multiple words can be represented by computing a resultant vector of the individual word vectors in the text. Moreover, to compute the similarity/relatedness between two words (texts), we can compute the cosine similarities between the corresponding vectors.

Since different approaches have different assumptions, it is hoped that they capture different aspects of a word's meaning. Thus, it can be expected that combining individual representations complements the coverage of the semantic aspects of a word and thus better represents the word than the individual representations. Nevertheless, how to combine heterogeneous models that have different underlying assumptions has not been explored much. This paper proposes different strategies for combining such representations and reports the performances on standard datasets.

The paper is organized as follows. In Chapter 2 we present the related works. In Chapter 3 and Chapter 4, we describe the popular representation techniques and some approaches to combination them respectively. In Chapter 5 we present the experiments and discuss the results obtained from the experiments. In Chapter 6 we conclude the findings.

Related Works

The literature for computing word-to-word similarity and relatedness is very rich. Broadly, these methods can be categorized into three groups depending on the type of resources they use: Knowledge-based, Corpus-based and Web-based. Knowledge-based methods rely on some form of ontology. WordNet (Miller 1995) is a well-known knowledge source that has been widely used to compute the semantic similarity and relatedness between words. It is a large lexical database of English consisting of nouns, verbs, adjectives and adverbs that are grouped into concepts i.e. synsets (synonym sets). The concepts are then linked through lexico-semantic relations such as hypernymy (is-a type of relation). The graph of lexicons has been exploited in different ways resulting in several similarity measures (Lin 1998; Hirst and St-Onge 1998; Wu and Palmer 1994; Banerjee and Pedersen 2003).

Corpus-based measures compute word similarity / relatedness scores based on the words' representations obtained from a given corpus. LDA, LSA and Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch 2007) are some of the most popular approaches for inferring word representations based on which a number of approaches have been devised (Rus et al. 2013). Most recently, neural models have been proposed to derive word representations from a corpus (Mikolov et al. 2013; Turian, Ratinov, and Bengio 2010). These measures have diverse assumptions and range from algebraic to probabilistic methods. Since we are going to combine these methods, we give a more detailed account of these approaches in the next chapter.

A Web-based approach relies on the web search results produced by search engines corresponding to supplied queries. Point-wise Mutual Information (PMI) (Bollegala, Matsuo, and Ishizuka 2007) is a famous example of this category. It gathers co-occurrence statistics based on the search engine results and uses that to compute word relatedness.

The plethora of measures available in the literature suggests that no single method is capable of adequately quantifying the similarity/relatedness between words. Therefore, combining different approaches may provide a better result. In fact, Stefuanescu et al. (2014) already hinted at a potential benefit of combining different approaches. Work by Yih and Qazvinian (2012) has already shown the effectiveness of combining vector space models of the same type for word relatedness measures. However, combining heterogeneous models that have different underlying assumptions and semantic spaces has not been studied much. This paper is a step in this direction.

Word Representation Approaches

Several methods are available in the literature that automatically learn word representation from a text corpus. These methods range from algebraic to probabilistic. Some of the examples include distributional approaches like Latent Semantic Analysis, Latent Dirichlet Allocation, Vector Space Model (VSM) and Explicit Semantic Analysis (ESA), and distributed word representation approaches such as by Mikolov et al. (2013). We briefly describe below the methods that we are going to use in our experiment.

LSA: Latent Semantic Analysis (LSA) is an algebraic method that represents the meaning of words as a vector in multi-dimensional semantic space (Landauer et al. 2007). LSA starts by creating a word-document matrix. It then applies singular value decomposition of the matrix followed by the factor-analysis. Usually around 300 factors (i.e. the number of dimensions) are kept, reducing significantly the original space. In other words, a word is a point in the new semantic space. Semantically similar words appear to be closer in the reduced space.

LDA : Latent Dirichlet Allocation (LDA) is a probabilistic topic model (Blei, Ng, and Jordan 2003). A topic model assumes that documents are distributions over a set of topics. The topics are distributions over words. Each word belongs to each topic with separate probability scores. That is, for a topic (say *Sports*) some words appear with higher probabilities (e.g. *tennis* and *football*) than other words (e.g. *food* and *poem*). Similarly, for a topic like *Food*, words like *sushi* and *tasty* have higher probabilities than words like *tennis*, *poem*. Note that LDA does not explicitly provide labels for each topic it generates. The names in the example above are for illustration purpose only.

Given a corpus, LDA automatically captures the topic mixtures for documents, and probabilities of words for each topic. The number of topics and hyper-parameters have to be specified.

Since a word appears in different topics with separate probability scores, we represent a word by a vector of length T where T is the number of topics. Moreover, if we consider a topic as a sense, LDA can capture polysemy, i.e. the property of a word to have multiple meanings, which is different from LSA because in LSA each word has a unique vector representation. This is a motivation for us to combine different models as they are capable of capturing different aspects of a word.

Deep Learning / Distributed Vector Representation : Deep Learning methods learn the distributed representations of concepts (usually called word embeddings). The distributed representation is one in which each entity is represented by a pattern of activities distributed over many computing elements, and each computing element is involved in representing many different entities (Hinton 1984). Specifically, in neural network based models, each concept is represented by many neurons and each neuron participated in the representation of many concepts. The patterns of activity across a number of units densely represent the meaning of concepts. Collobert and Weston (2008) used Convolutional Neural Network architecture to learn word embeddings and applied them for multiple NLP predictions. One of the recent works on distributed representations is by Mikolov et al. (2013) where they used probabilistic feed forward neural network language model to estimate word representations in vector space.

Once the vector representations for words are obtained, the similarity between two words can be easily computed. For instance, to compute the similarity between word W_i and word W_j, we use the cosine similarity between word vectors as :

$Sim(W_i, W_j) = \frac{\sum_{n=1}^{K} V_i[n] * V_j[n]}{|V_i| * |V_j|}$ where, V_i and V_j are the vectors corresponding to word W_i and W_j respectively, and K is the dimension of vector V_i (= V_j).

Combining Word Representations

We believe that each aforementioned word representation method represents different aspects of a word's meaning since they have different assumptions. This motivates us to combine individual representation with the hope of getting more coverage of semantic aspects of a word. This hopefully better represents the word than the individual one. We describe below the two approaches for combining word representations.

A. Extend: In this method, we append individual vectors and create a new vector. Mathematically, given M vectors $V_1...V_M$ with respective dimensions $d_1, d_2, ..., d_n$, we construct a single vector V as follows:

$$V[i] = \begin{cases} V_1[i] & \text{if } 0 \leq i < d_1 \\ V_2[i - d_1] & \text{if } d_1 \leq i < d_1 + d_2 \\ . & . \\ . & . \\ V_M[i - \sum_{j=1}^{M-1} d_j] & \text{if } \sum_{j=1}^{M-1} d_j \leq i < \sum_{j=1}^{M} d_j \end{cases}$$

B. Average: This method computes semantic similarity scores for each model and then takes the mean score as the score predicted by the system.

One crucial point is about the scaling of the vectors obtained from the different semantic spaces. Since each semantic space has different assumptions, the vectors from these spaces have different scales. Thus, it might be effective to normalize them before applying the *Extend* technique. We present the effect of vector scaling later in the experiment section. It is important to note, however, that mathematically the aforementioned approaches for combining word representations (i.e. *Extend* and *Average*) would be the same if the individual vector is a unit vector. In other words, extending unit vectors and computing a cosine similarity using the combined vector is equivalent to averaging of cosine similarities from the individual vectors.

Policy for handling missing vectors : It is possible that some words may not have their vector representations in a given model. This may happen either because the model generation process is expensive and thus some of the words have to be removed or the corpus from which the model was generated might not contain the words or something else. In those situations, we represent the word by one of its synonyms, extracted from the WordNet, that is present in the model.

Experiments and Results

We selected six popular word representation models and then evaluated them against two standard datasets.

Selected Models

LSA$_{TASA}$: It is the LSA space generated from the TASA corpus (compiled by Touchstone Applied Science Associates). The corpus is a balanced collection of 60,527 samples from 6333 textbooks and covers various genres such as science, language arts, health, economics, social studies, business, and others.

LSA$_{Wiki}$: We used the LSA model (Wiki_NVAR_f7) generated from Wikipedia by Stefanescu, Banjade, and Rus (2014)[1]. The model was generated by considering only the lemma of the content words that appeared at least 7 times in the corpus.

LDA$_{Wiki}$: To generate the LDA model from Wikipedia, we filtered out the documents that have less than 500 words, the words that have less than 500 entries, and the stop words. This gave us 270290 documents and the vocabulary of 59136 words. With this data, we generated a 300 topic LDA model.

NLM model (Turian): We used pre-trained Neural Language Model (NLM) vector model generated by Turian, Ratinov, and Bengio (2010)[2]. In this representation, each distributed word representation consisted of 200 dimensions and were induced on the large unlabeled RCV1 corpus (about 37M words of Reuter News Text) in a general and unsupervised manner.

NLM model (Mikolov): This model is a pre-trained vector model based on Google News dataset (about 100 billion words) and is prepared by Mikolov et al.(2013). The distributed word vectors were computed using skip-gram model. The model contains 300-dimensional vectors for 3 million words and phrases[3].

[1] http://www.semanticsimilarity.org/

[2] http://metaoptimize.com/projects/wordreprs/

[3] https://code.google.com/p/word2vec/

Table 1: *Performance of different models in SimLex-999 and Word-Sim353 datasets*

	LSA$_{wiki}$	**LSA**$_{Tasa}$	**LDA**$_{wiki}$	**NLM**$_{Turian}$	**NLM**$_{Mikolov}$	**MDL**$_{GloVe}$
SimLex-999	0.27	0.27	0.29	0.16	**0.44**	0.37
Word-Sim353	0.59	0.54	0.65	0.26	**0.68**	0.63

GloVe Model: GloVe (Global Vector) is an unsupervised learning model for word representation (Jeffrey, Socher, and Manning 2014). The model is trained on the non-zero elements in a global word-word co-occurrence matrix. We used the pre-trained model GloVe-42B which was trained on 42 billion words[4].

Benchmark Datasets

To evaluate the system, we followed the standard approach in which the Spearman's rank correlation coefficient is computed between the scores produced by the system and the human judgments on a set of word pairs. We used two such datasets : WordSim-353 and Simlex-999.

WordSim-353 : This is the largest dataset that has been used extensively to evaluate the word relatedness measures. It is prepared by Finkelstein et al. (2001) and contains of 353 word pairs. Each word pair was scored by 13-16 judges on a scale of 0-10. The mean score of all the judges is taken as the actual human score and used to evaluate the proposed methods.

Simlex-999 : The recent, largest dataset available to evaluate the word similarity, as opposed to word relatedness, is prepared by Hill, Reichart, and Korhonen (2014) . It consists of a set of 999 word pairs. It is claimed as a balanced dataset as the pairs include 666 noun, 222 verb and 111 adjective pairs. Each pair is scored by at least 36 native English speakers. The mean score is considered the final score for human judgment and used to compare against the proposed methods.

Evaluations

It would be interesting to see how these different representations perform on the relatedness and similarity measures individually. For this, we evaluated the individual models on the *Simlex-999* and *WordSim-353* datasets. The results are presented in Table 1. NLM$_{Mikolov}$ model has the best and NLM$_{Turian}$ has the least Spearman's rank correlation with the human judgment for both the relatedness and similarity measures. The findings are consistent with that reported by Hill, Reichart, and Korhonen (2014). NLM$_{Mikolov}$ leads the other measures with a wide margin for similarity measure but for relatedness measure LDA$_{wiki}$ and MDL$_{GloVe}$ are very competitive. The higher scores of the six models on *WordSim-353* compared to *Simlex-999* indicate that both the distributional(LSA, LDA, and MDL$_{GloVe}$) and distributed methods (NLMs) capture word relatedness better than word similarities.

Effect of normalization : As mentioned previously, the vectors obtained from different models can have different scales. We wanted to see what and how much effect would it have when we use raw vectors (unnormalized) instead of unit vectors (i.e. normalized vectors). We present the performances for some combination of models using both the *Extend* and *Average* in Table 2 since, as mentioned in previous section, *Extend* and *Average* would yield different results when raw vectors are used. As we can see, using raw vectors resulted in very poor performance compared to when using the corresponding unit vectors. This observation was consistent for the rest of the combinations as well as for the similarity measures (i.e. in *Simlex-999* dataset). Thus, for the rest of the experiments, we used normalized vectors and thus reported only average scores.

Table 2: *Effect of normalization of vectors in measuring Word relatedness (WordSim-353)*

Methods	Extend	Average
LDA$_{wiki}$ + NLM$_{Mikolov}$	0.682	0.739
LDA$_{wiki}$ + MDL$_{GloVe}$	0.632	0.714
LDA$_{wiki}$ + LSA$_{wiki}$	0.592	0.657

Effect of combination : To answer the question of whether combining different representations would be productive for measuring word relatedness, we evaluated all the combinations of 6 models (total 63) in *WordSim-353* dataset. We reported the top five best performing combinations as well as some other interesting cases in Table 3. The combination of LDA$_{wiki}$, NLM$_{Mikolov}$, and MDL$_{GloVe}$ outperformed the rest combinations with the correlation score of 0.757 with human judges. It is better than the individual performance (see Table 1). It performs even better than combining all methods (*All* in the Table 3). This might be because some of the low performing models (e.g. NLM$_{Turian}$) affected the relatedness score while averaging them. The observation that the best combination (i.e. LDA$_{wiki}$ + NLM$_{Mikolov}$ + MDL$_{GloVe}$) included the distributional and distributed (neural language model) models suggests that word relatedness measure could be improved by combining diverse representations.

To compare the performances of the combined representations for relatedness with the existing methods, we collected the performances of the existing methods from the literature for the same dataset (*WordSim-353*) and reported them in the bottom part of Table 3. Yih and Qazvinian (2012) reported the highest correlation of 0.81. Except that, our top performing combinations are superior if not as competitive as most of the methods reported in the literature.

Similarly, to see if combining these models can improve word similarity measure, we evaluated all the combinations

[4] http://www-nlp.stanford.edu/projects/glove/

Table 3: *Performance (Spearman's rank correlation) of different combination of methods in WordSim-353 dataset (to measure word relatedness)*

Methods	Average
LDA$_{wiki}$ + NLM$_{Mikolov}$ + MDL$_{GloVe}$	**0.757**
LDA$_{wiki}$ + NLM$_{Mikolov}$ + MDL$_{GloVe}$ + LSA$_{wiki}$	0.747
LDA$_{wiki}$ + NLM$_{Mikolov}$ + MDL$_{GloVe}$ + LSA$_{tasa}$	0.743
LDA$_{wiki}$ + NLM$_{Mikolov}$	0.739
LSA$_{wiki}$ + NLM$_{Mikolov}$ + MDL$_{GloVe}$	0.735
All	0.724
NLM$_{Mikolov}$ + MDL$_{GloVe}$	0.717
LDA$_{wiki}$ + MDL$_{GloVe}$	0.714
LSA$_{wiki}$ + MDL$_{GloVe}$	0.690
LDA$_{wiki}$ + LSA$_{wiki}$	0.657
Yih and Qazvinian (2012)	0.810
Gabrilovich and Markovitch (2007)	0.748
Hassan and Mihalcea (2011):Multi-lingual SSA	0.713
Luong et al. (2013)	0.650
Hassan and Mihalcea (2011):SSA	0.622
Collobert and Weston (2008)	0.500
Hassan and Mihalcea (2011):Resnik	0.353
Hassan and Mihalcea (2011):Lin	0.348

Table 4: *Performance (Spearman's rank correlation) of different combination of methods in Simlex-999 dataset (to measure word similarity)*

Methods	Average
LDA$_{Mikolov}$ + MDL$_{GloVe}$	0.435
LDA$_{Mikolov}$ + MDL$_{GloVe}$ + LSA$_{Tasa}$	0.426
LDA$_{Mikolov}$ + MDL$_{GloVe}$ + LSA$_{Tasa}$+ NLM$_{Turian}$	0.415
All	0.385

(total 63) of the models against *Simlex-999* dataset. Results for the top three best performing combinations as well as the combination of all the methods (designated by *All*) are presented in Table 4. The combination of NLM$_{Mikolov}$ and NLM$_{GloVe}$ achieved the highest correlation score of 0.435. This is even less than the performance of NLM$_{Mikolov}$ alone (see Table 1). Thus, the combination of distributed and distributional models does not seem to be useful for word similarities as compared to the word relatedness. This could be because all these models rely on distributional hypothesis according to which words that are used and occur in the same contexts tend to contain similar meanings. The hypothesis is blamed for scoring higher for related words than the synonyms (Yih and Qazvinian 2012; Han et al. 2013). As a result, such methods most likely give higher scores for (*chicken*, *egg*) than (*chicken* , *hen*) due to abundance of the phrase *chicken and egg* in the corpus compared to *chicken and hen*. Thus, assistance from Knowledge-based approach is a must if we want to improve the similarity measures. In fact, Han et al. (2013) have already succeeded to boost the similarity measures by combining rules from WordNet and the LSA model.

Conclusions

In this paper, we sought to find if combination of heterogeneous word representations are helpful for measuring word relatedness and similarity. Particularly, we chose six well-known distributional and distributed models and proposed methods to combine the word vectors. Our experiments showed that :

- Given the diverse semantic spaces, normalization of vectors must be done before exploiting them.
- Combining word representations from distributional and distributed models is not sufficient for improving word similarity measure. Thus, incorporation of Knowledge-based resources/approaches is recommended to improve word similarity measures.
- Combining word representations from distributional and distributed models, however, can improve the word relatedness measure.

As a future work, we want to address the weaknesses of the distributional hypothesis e.g. by incorporating the features from ontologies to improve the similarity measures.

Acknowledgments

This research was partially sponsored by The University of Memphis and the Institute for Education Sciences (IES) under award R305A100875 to Dr. Vasile Rus.

References

Banerjee, S., and Pedersen, T. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *IJCAI*, volume 3, 805–810.

Bayardo, R. J.; Ma, Y.; and Srikant, R. 2007. Scaling up all pairs similarity search. In *Proceedings of the 16th international conference on World Wide Web*, 131–140. ACM.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *the Journal of machine Learning research* 3:993–1022.

Bollegala, D.; Matsuo, Y.; and Ishizuka, M. 2007. Measuring semantic similarity between words using web search engines. *www* 7:757–766.

Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, 160–167. ACM.

Dagan, I.; Glickman, O.; and Magnini, B. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*. Springer. 177–190.

Finkelstein, L.; Gabrilovich, E.; Matias, Y.; Rivlin, E.; Solan, Z.; Wolfman, G.; and Ruppin, E. 2001. Placing search in context: The concept revisited. In *Proceedings of* *the 10th international conference on World Wide Web*, 406–414. ACM.

Gabrilovich, E., and Markovitch, S. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, 1606–1611.

Han, L.; Kashyap, A.; Mayfield, T. F. J.; and Weese, J. 2013. Umbc_ebiquity-core: Semantic textual similarity systems. *Atlanta, Georgia, USA* 44.

Hassan, S., and Mihalcea, R. 2011. Semantic relatedness using salient semantic analysis. In *AAAI*.

Hill, F.; Reichart, R.; and Korhonen, A. 2014. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *arXiv preprint arXiv:1408.3456*.

Hinton, G. E. 1984. Distributed representations.

Hirst, G., and St-Onge, D. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database* 305:305–332.

Hliaoutakis, A.; Varelas, G.; Voutsakis, E.; Petrakis, E. G.; and Milios, E. 2006. Information retrieval by semantic similarity. *International journal on semantic Web and information systems (IJSWIS)* 2(3):55–73.

Jeffrey, P.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. 1532–1543.

Landauer, T. K.; McNamara, D. S.; Dennis, S.; and Kintsch, W. 2007. *Handbook of latent semantic analysis*. Psychology Press.

Lin, D. 1998. An information-theoretic definition of similarity. In *ICML*, volume 98, 296–304.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 3111–3119.

Miller, G. A. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Niraula, N.; Banjade, R.; Ştefănescu, D.; and Rus, V. 2013. Experiments with semantic similarity measures based on lda and lsa. In *Statistical Language and Speech Processing*, 188–199. Springer.

Osman, A. H.; Salim, N.; Binwahlan, M. S.; Alteeb, R.; and Abuobieda, A. 2012. An improved plagiarism detection scheme based on semantic role labeling. *Applied Soft Computing* 12(5):1493–1502.

Rus, V., and Graesser, A. C. 2006. Deeper natural language processing for evaluating student answers in intelligent tutoring systems. In *Proceedings of the National Conference on Artificial INtelligence*, volume 21, 1495. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Rus, V., and Lintean, M. 2012. A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, 157–162. Association for Computational Linguistics.

Rus, V.; Lintean, M. C.; Banjade, R.; Niraula, N. B.; and Stefanescu, D. 2013. Semilar: The semantic similarity toolkit. In *ACL (Conference System Demonstrations)*, 163–168. Association for Computational Linguistics.

Socher, R.; Huang, E. H.; Pennin, J.; Manning, C. D.; and Ng, A. Y. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*, 801–809.

Stefanescu, D.; Banjade, R.; and Rus, V. 2014. Latent semantic analysis models on wikipedia and tasa. The 9th Language Resources and Evaluation Conference (LREC 2014).

Stefuanescu, D.; Rus, V.; Niraula, N. B.; and Banjade, R. 2014. Combining knowledge and corpus-b to-word similarity measures for word.

Turian, J.; Ratinov, L.; and Bengio, Y. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 384–394. Association for Computational Linguistics.

Wu, Z., and Palmer, M. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, 133–138. Association for Computational Linguistics.

Yih, W.-t., and Qazvinian, V. 2012. Measuring word relatedness using heterogeneous vector space models. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 616–620. Association for Computational Linguistics.