NeRoSim: A System for Measuring and Interpreting Semantic Textual Similarity

Rajendra Banjade[*], Nobal B. Niraula[*], Nabin Maharjan[*], Vasile Rus, Dan Stefanescu[†], Mihai Lintean[†], Dipesh Gautam
Department of Computer Science
The University of Memphis
Memphis, TN
{rbanjade,nbnraula,nmharjan,vrus,dstfnscu,mclinten,dgautam}@memphis.edu

Abstract

We present in this paper our system developed for SemEval 2015 Shared Task 2 (2a - English Semantic Textual Similarity, STS, and 2c - Interpretable Similarity) and the results of the submitted runs. For the English STS subtask, we used regression models combining a wide array of features including semantic similarity scores obtained from various methods. One of our runs achieved weighted mean correlation score of 0.784 for sentence similarity subtask (i.e., English STS) and was ranked tenth among 74 runs submitted by 29 teams. For the interpretable similarity pilot task, we employed a rule-based approach blended with chunk alignment labeling and scoring based on semantic similarity features. Our system for interpretable text similarity was among the top three best performing systems.

1 Introduction

Semantic Textual Similarity (STS) is the task of measuring the degree of semantic equivalence for a given pair of texts. The importance of semantic similarity in Natural Language Processing is highlighted by the diversity of datasets and shared task evaluation campaigns over the last decade (Dolan et al., 2004; Agirre et al., 2012; Agirre et al., 2013; Agirre et al., 2014; Rus et al., 2014) and by many uses such as in text summarization (Aliguliyev, 2009) and student answer assessment (Rus and Lintean, 2012; Niraula et al., 2013).

This year's SemEval shared task on semantic textual similarity focused on English STS, Spanish STS, and Interpretable Similarity (Agirre et al., 2015). We participated in the English STS and Interpretable Similarity subtasks. We describe in this paper our systems participated in these two subtasks.

The English STS subtask was about assigning a similarity score between 0 and 5 to pairs of sentences; a score of 0 meaning the sentences are unrelated and 5 indicating they are equivalent. Our three runs for this subtask combined a wide array of features including similarity scores calculated using knowledge based and corpus based methods in a regression model (cf. Section 2). One of our systems achieved mean correlation score of 0.784 with human judgment on the test data.

Although STS systems measure the degree of semantic equivalence in terms of a score which is useful in many tasks, they stop short of explaining why the texts are similar, related, or unrelated. They do not indicate what kind of semantic relations exist among the constituents (words or chunks) of the target texts. Finding explicit relations between constituents in the paired texts would enable a meaningful interpretation of the similarity scores. To this end, Brockett (2007) and Rus et al. (2012) produced datasets where corresponding words (or multiword expressions) were aligned and in the later case their semantic relations were explicitly labeled. Similarly, this year's pilot subtask called Interpretable Similarity required systems to align the segments (chunks) either using the chunked texts given by the organizers or chunking the given texts and indicating the type of semantic relations (such as EQUI for

[*] These authors contributed equally to this work
[†] Work done while at University of Memphis

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 164–171,
Denver, Colorado, June 4-5, 2015.

equivalent, OPPO for opposite) between each pair of aligned chunks. Moreover, a similarity score for each alignment ($0 -$ unrelated, $5 -$ equivalent) had to be assigned. We applied a set of rules blended with similarity features in order to assign the labels and scores for the chunk-level relations (cf. Section 3). Our system was among the top performing systems in this subtask.

2 System for English STS

We used regression models to compute final sentence-to-sentence similarity scores using various features such as different sentence-to-sentence similarity scores, presence of negation cues, lexical overlap measures etc. The sentence-to-sentence similarity scores were calculated using word-to-word similarity methods and optimal word and chunk alignments.

2.1 Word-to-Word Similarity

We used knowledge based, corpus based, and hybrid methods to compute word-to-word similarity. From the knowledge based category, we used WordNet (Fellbaum, 1998) based similarity methods from SEMILAR Toolkit (Rus et al., 2013) which include Lin (Lin, 1998), Lesk (Banerjee and Pedersen, 2003), Hso (Hirst and St-Onge, 1998), Jcn (Jiang and Conrath, 1997), Res (Resnik, 1995), Path, Lch (Leacock and Chodorow, 1998), and Wup (Wu and Palmer, 1994).

In corpus based category, we developed Latent Semantic Analysis (LSA) (Landauer et al., 2007) models[1] from the whole Wikipedia articles as described in Stefanescu et al. (2014a). We also used pre-trained Mikolov word representations (Mikolov et al., 2013)[2] and GloVe word vectors (Pennington et al., 2014)[3]. In these cases, each word was represented as a vector encoding and the similarity between words were computed as cosine similarity between corresponding vectors. We exploited the lexical relations between words, i.e. synonymy and antonymy, from WordNet 3.0. As such we computed similarity scores between two words a and b as:

$$sim(a,b) = \begin{cases} 1, \text{ if a and b are synonyms} \\ 0, \text{ if a and b are antonyms} \\ \dfrac{\mathbf{A}.\mathbf{B}}{|\mathbf{A}||\mathbf{B}|}, \text{ otherwise} \end{cases}$$

where $\mathbf{A}$ and $\mathbf{B}$ are vector representations of words a and b respectively.

In hybrid approach, we developed a new word-to-word similarity measure (hereafter referred as Combined-Word-Measure) by combining the WordNet-based similarity methods with corpus based methods (using Mikolov's word embeddings and GloVe vectors) by applying Support Vector Regression (Banjade et al., 2015).

2.2 Sentence-to-Sentence Similarity

We applied three different approaches to compute sentence-to-sentence similarity.

2.2.1 Optimal Word Alignment Method

Our alignment step was based on the optimal assignment problem, a fundamental combinatorial optimization problem which consists of finding a maximum weight matching in a weighted bipartite graph. An algorithm, the Kuhn-Munkres method (Kuhn, 1955), can find solutions to the optimum assignment problem in polynomial time.

In our case, we first computed the similarity of word pairs (all possible combinations) using all similarity methods described in Section 2.1. The similarity score less than 0.3 (empirically set threshold), was reset to 0 in order to avoid noisy alignments. Then the words were aligned so that the overall alignment score between the full sentences was maximum. Once the words were aligned optimally, we calculated the sentence similarity score as the sum of the word alignment scores normalized by the average length of the sentence pair.

2.2.2 Optimal Chunk Alignment Method

We created chunks and aligned them to calculate sentence similarity as in Stefanescu et al. (2014b) and applied optimal alignment twice. First, we applied optimal alignment of words in two chunks to measure the similarity of the chunks. As before, word similarity threshold was set to 0.3. We then

[1]Models available at http://semanticsimilarity.org
[2]Downloaded from http://code.google.com/p/word2vec/
[3]Downloaded from http://nlp.stanford.edu/projects/glove/

normalized chunk similarity by the number of tokens in the shorter chunk such that it assigned higher scores to pairs of chunks such as *physician* and *general physician*. Second, we applied optimal alignment at chunk level in order to calculate the sentence level similarity. We used chunk-to-chunk similarity threshold 0.4 to prevent noisy alignments. In this case, however, the similarity score was normalized by the average number of chunks in the given texts pair. All threshold values were set empirically based on the performance on the training set.

2.2.3 Resultant Vector Based Method

In this approach, we combined vector based word representations to obtain sentence level representations through vector algebra. We added the vectors corresponding to content words in each sentence to create a resultant vector for each sentence and the cosine similarity was calculated between the resultant vectors. We used word vector representations from Wiki LSA, Mikolov and GloVe models.

For a missing word, we used vector representation of one of its synonyms obtained from the WordNet. To compute the synonym list, we considered all senses of the missing word given its POS category.

2.3 Features for Regression

We summarize the features used for regression next.

1. Similarity scores using optimal alignment of words where word-to-word similarity was calculated using vector based methods using word representations from Mikolov, GloVe, LSA Wiki models and Combined-Word-Measure which combines knowledge based methods and corpus based methods.

2. Similarity score using optimal alignment of chunks where word-to-word similarity scores were calculated using Mikolov's word representations.

3. Similarity scores based on the resultant vector method using word representations from Mikolov, GloVe, and LSA Wiki models.

4. Noun-Noun, Adjective-Adjective, Adverb-Adverb, and Verb-Verb similarity scores and similarity score for other words using optimal word alignment and Mikolov's word representations.

5. Multiplication of noun-noun similarity score and verb-verb similarity score (scores calculated as described in 4).

6. Whether there was any antonym pair present.

7. $\frac{|C_{i1} - C_{i2}|}{C_{i1} + C_{i2}}$ where C_{i1} and C_{i2} are the counts of $i \in \{$all tokens, adjectives, adverbs, nouns, and verbs$\}$ for sentence 1 and 2 respectively.

8. Presence of adjectives and adverbs in first sentence, and in the second sentence.

9. Unigram overlap with synonym check, bigram overlap and BLEU score (Papineni et al., 2002).

10. Presence of negation cue (e.g. no, not, never) in either of sentences.

11. Whether one sentence was a question while the other was not.

12. Total number of words in each sentence. Similarly, the number of adjectives, nouns, verbs, adverbs, and others, in each sentence.

Data set	Count	Release time
SMTnews	351	STS2012-Test
Headlines	1500	STS2013-Test
Deft-forum	423	STS2014-Test
Deft-news	299	STS2014-Test
Images	749	STS2014-Test

Table 1: Summary of training data

2.4 Experiments and Results

Data: For training, we used data released in previous shared tasks (summarized in Table 1). We selected datasets that included texts from different genres. However, some others, such as Tweet-news and MSRPar were not included. Tweet-news data were quite different from most other texts. MSRPar, being more biased towards overlapping text (Rus et al., 2014), was also a concern.

The test set included data (sentence pairs) from Answers-forums (375), Answers-students (750), Belief (375), Headlines (750), and Images (750).

Preprocessing: We removed stop words, labeled each word with Part-of-Speech (POS) tag and lemmatized them using Stanford CoreNLP Toolkit (Manning et al., 2014). We did spelling corrections in student answers and forum data using Jazzy tool (Idzelis, 2005) with WordNet dictionary. Moreover, in student answers data, we found that the symbol A (such as in bulb A and node A) typed in lowercase was incorrectly labeled as a determiner 'a' by the POS tagger. We applied a rule to correct it. If the token after 'a' is not an adjective, adverb, or noun, or the token is the last token in the sentence, we changed its type to noun (NN). We then created chunks as described by Stefuanescu et al. (2014b).

Regression: We generated various features as described in Section 2.3 and applied regression methods in three different settings. In the first run (R1), all features were used in Support Vector Regression (SVR) with Radial Basis Function kernel. The second run (R2) was same as R1 except that the features in R2 did not include the count features (i.e., features in 12). In the third run (R3), we used features same as R2 but applied linear regression instead.

For SVR, we used LibSVM library (Chang and Lin, 2011) in Weka (Holmes et al., 1994) and for the linear regression we used Weka's implementation. The 10-fold cross validation results (r) of three different runs with the training data were 0.7734 (R1), 0.7662 (R2), and 0.7654 (R3).

Data set	Baseline	R1	R2	R3
Ans-forums	0.445	0.526	0.694	0.677
Ans-students	0.664	0.725	0.744	0.735
Belief	0.651	0.631	0.751	0.722
Headlines	0.531	0.813	0.807	0.812
Images	0.603	0.858	0.864	0.857
Mean	0.587	0.743	**0.784**	0.776

Table 2: Results of our submitted runs on test data.

The results on the test set have been presented in Table 2. Though R1 had the highest correlation score in a 10-fold cross validation process using the training data, the results of R2 and R3 on the test data were consistently better than the results of R1. It suggests that absolute count features used in R1 tend to overfit the model. The weighted mean correlation of R2 was 0.784 - the best among our three runs and ranked 10th among 74 runs submitted by 29

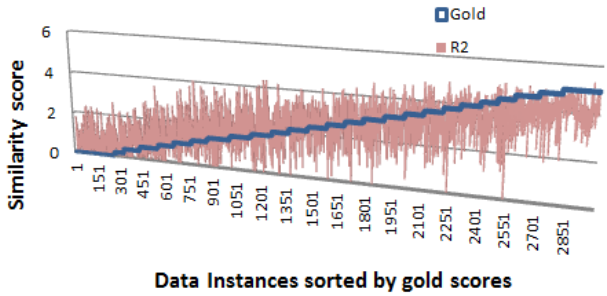

Figure 1: A graph showing similarity scores predicted by our system (R2) and corresponding human judgment in test data (sorted by gold score).

participating teams. The correlation score was very close to the results of other best performing systems. Moreover, we observed from Figure 1 that our system worked fairly well at all range of scores. The actual variation of scores at extreme (very low and very high) points is not very high though the regression line seems to be more skewed. However, the correlation scores of answer-forum, answer-students, and belief data were found to be lower than those of headlines and images data. The reason might be the texts in the former data being not well-written as compared to the latter. Also, more contextual information is required to fully understand them.

3 Interpretable STS

For each sentence pair, participating systems had to identify the chunks in each sentence or use the given gold chunks, align corresponding chunks and assign a similarity/relatedness score and type of the alignment for each alignment. The alignment types were EQUI (semantically equivalent), OPPO (opposite in meaning), SPE (one chunk is more specific than other), SIM (similar meanings, but no EQUI, OPPO, SPE), REL (related meanings, but no SIM, EQUI, OPPO, SPE), ALIC (does not have any corresponding chunk in the other sentence because of the 1:1 alignment restriction), and NOALI (has no corresponding chunk in the other sentence). Further details about the task including type of relations and evaluation criteria can be found in Agirre et al. (2015).

Our system uses gold chunks of a given sentence pair and maps chunks of the first sentence to those

from the second by assigning different relations and scores based on a set of rules. The system performs stop word marking, POS tagging, lemmatization, and named-entity recognition in the preprocessing steps. It also uses lookups for synonym, antonym and hypernym relations.

For synonym lookup, we created a strict synonym lookup file using WordNet. Similarly, an antonym lookup file was created by building an antonym set for a given word from its direct antonyms and their synsets. We further constructed another lookup file for strict hypernyms.

3.1 Rules

In this section, we describe the rules used for chunk alignments and scoring. The scores given by each rule are highlighted.

Conditions: We define below a number of conditions for a given chunk pair that might be checked before applying a rule.

C_1: One chunk has a conjunction and other does not

C_2: A content word in a chunk has an antonym in the other chunk

C_3: A word in either chunk is a NUMERIC entity

C_4: Both chunks have LOCATION entities

C_5: Any of the chunks has a DATE/TIME entity

C_6: Both chunks share at least one content word other than noun

C_7: Any of the chunks has a conjunction

Next, we define a set of rules for each relation type. For aligning a chunk pair (A, B), these rules are applied in order of precedence as NOALIC, EQUI, OPPO, SPE, SIMI, REL, and ALIC. Once a chunk is aligned, it would not be considered for further alignments. Moreover, there is a precedence of rules within each relation type e.g. EQ_2 is applied only if EQ_1 fails and EQ_3 is applied if both EQ_1 and EQ_2 fail and so on. If a chunk does not get any relation after applying all the rules, a NOALIC relation is assigned. Note that we frequently use $sim\text{-}Mikolov(A, B)$ to refer to the similarity score between the chunks A and B using Mikolov word vectors as described in Section 2.2.2.

3.1.1 NOALIC Rules

NO_1: If a chunk to be mapped is a single token and is a punctuation, assign NOALIC.

3.1.2 EQUI Rules

EQUI Rules $EQ_1 - EQ_3$ are applied unconditionally. The rest rules ($EQ_4 - EQ_5$) are applied only if none of conditions C_1 - C_5 are satisfied.

EQ_1 - Both chunks have same tokens (**5**) - e.g. to compete ⇔ To Compete

EQ_2 - Both chunks have same content words (**5**) - e.g. in Olympics ⇔ At Olympics

EQ_3 - All content words match using synonym lookup (**5**) - e.g. to permit ⇔ Allowed

EQ_4 : All content words of a chunk match and unmatched content word(s) of the other chunk are all of proper noun type (**5**) - e.g. Boeing 787 Dreamliner ⇔ on 787 Dreamliner

EQ_5 : Both chunks have equal number of content words and $sim - Mikolov(A, B) > 0.6$ (**5**) - e.g. in Indonesia boat sinking ⇔ in Indonesia boat capsize

3.1.3 OPPO Rules

OPPO rules are applied only when none of C_3 and C_7 are satisfied.

OP_1: A content word in a chunk has an antonym in the other chunk (**4**) - e.g. in southern Iraq ⇔ in northern Iraq

3.1.4 SPE Rules

SP_1: If chunk A but B has a conjunction and A contains all the content words of B then A is SPE of B (**4**) - e.g. Angelina Jolie ⇔ Angelina Jolie and the complex truth.

SP_2: If chunk A contains all content words of chunk B plus some extra content words that are not verbs, A is a SPE of B or vice-versa. If chunk B has multiple SPEs, then the chunk with the maximum token overlap with B is selected as the SPE of B. (**4**) - e.g. Blade Runner Pistorius ⇔ Pistorius.

SP_3: If chunks A and B contain only one noun each say n_1 and n_2 and n_1 is hypernym of n_2, B is SPE of A or vice versa (**4**) - e.g. by a shop ⇔ outside a bookstore.

3.1.5 SIMI Rules

SI_1: Only the unmatched content word in each chunk is a CD type(**3**)-e.g. 6.9 magnitude earthquake ⇔ 5.6 magnitude earthquake

SI_2: Each chunk has a token of DATE/TIME type (**3**)- e.g. on Friday ⇔ on Wednesday

	Run	A	T	S	T+S
Headlines	Baseline	0.844	0.555	0.755	0.555
	R_1	**0.898**	0.654	0.826	0.638
	R_2	0.897	0.655	**0.826**	0.640
	R_3	0.897	**0.666**	0.815	**0.642**
Images	Baseline	0.838	0.432	0.721	0.432
	R_1	**0.887**	**0.614**	**0.787**	**0.584**
	R_2	0.880	0.585	0.781	0.561
	R_3	0.883	0.603	0.783	0.575

Table 3: F_1 scores for Images and Headlines. A, T and S refer to Alignment, Type, and Score respectively. The highlighted scores are the best results produced by our system.

SI_3: Each chunk has a token of LOCATION type (**3**) - e.g. Syria $\Leftrightarrow$ Iraq
SI_4: When both chunks share at least one noun then assign **3** if $sim\text{-}Mikolov(A, B) >= 0.4$ and **2** otherwise. - e.g. Nato troops $\Leftrightarrow$ NATO strike
SI_5: This rule is applied only if C_6 is not satisfied. Scores are assigned as : (i) **4** if $sim\text{-}Mikolov(A, B) \in [0.7, 1.0]$ (ii) **3** if $sim\text{-}Mikolov(A, B) \in [0.65, 0.7)$ (iii) **2** if $sim\text{-}Mikolov(A, B) \in [0.60, 0.65)$

3.1.6 REL Rules

RE_1: If both chunks share at least one content word other than noun then assign REL relation. Scores are assigned as follows : (i) **4** if $sim\text{-}Mikolov(A, B) \in [0.5, 1.0]$ (ii) **3** if $sim\text{-}Mikolov(A, B) \in [0.4, 0.5)$ (iii) **2** otherwise. e.g. to Central African Republic $\Leftrightarrow$ in Central African capital

3.1.7 ALIC Rules

AL_1: If a chunk in a sentence X (C_x) is not aligned yet but has a chunk in another pair-sentence Y (C_y) that is already aligned and has $sim\text{-}Mikolov(C_x, C_y) >= 0.6$, assign ALIC relation to C_x with a score of (**0**).

3.2 Experiments and Results

We applied above mentioned rules in the training data set by varying thresholds for $sim\text{-}Mikolov$ scores and selected the thresholds that produced the best results in the training data set. Since three runs were allowed to submit, we defined them as follows:
$Run1(R_1)$: We applied our full set of rules with limited stop words (375 words). However EQ_4 was modified such that it would apply when unmatched content words of the bigger chunk were of noun rather than proper noun type.
$Run2(R_2)$: Same as R_1 but with extended stop words (686 words).
$Run3(R_3)$: Applied full set of rules with extended stop words.

The results corresponding to our three runs and that of the baseline are presented in Table 3. In Headlines test data, our system outperformed the rest competing submissions in all evaluation metrics (except when alignment type and score were ignored). In Images test data, R_1 was the best in alignment and type metrics. Our submissions were among the top performing submissions for score and type+score metrics.

R_3 performed better among all runs in case of Headlines data in overall. This was chiefly due to modified EQ_4 rule which reduced the number of incorrect EQUI alignments. We also observed that performance of our system was least affected by size of stopword list for Headlines data as both R_1 and R_2 recorded similar F_1-measures for all evaluation metrics. However, R_1 performed relatively better than R_2 in Images data-particularly in correctly aligning chunk relations. It could be that images are described mostly using common words and thus were filtered by R_2 as stop words.

4 Conclusion

In this paper we described our submissions to the Semantic Text Similarity Task in SemEval Shared Task 2015. Our system for the English STS subtask used regression models that combined a wide array of features including semantic similarity scores obtained with various methods. For the Interpretable Similarity subtask, we employed a rule-based approach for aligning chunks in sentence pairs and assigning relations and scores for the alignments. Our systems were among the top performing systems in both subtasks. We intend to publish our systems at http://semanticsimilarity.org.

Acknowledgments

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

References

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 385–393. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In *In* SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics*. Citeseer.

Eneko Agirre, Carmen Baneab, Claire Cardiec, Daniel Cerd, Mona Diabe, Aitor Gonzalez-Agirrea, Weiwei Guof, Rada Mihalceab, German Rigaua, and Janyce Wiebeg. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. *SemEval 2014*, page 81.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, CO, June. Association for Computational Linguistics.

Ramiz M Aliguliyev. 2009. A new sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Systems with Applications*, 36(4):7764–7772.

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

Rajendra Banjade, Nabin Maharjan, Nobal B. Niraula, Vasile Rus, and Dipesh Gautam. 2015. Lemon and tea are not similar: Measuring word-to-word similarity combining different methods. In *Proceedings of the 16th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 335–346.

Chris Brockett. 2007. Aligning the rte 2006 corpus. *Microsoft Research*.

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, page 350. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.

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

Geoffrey Holmes, Andrew Donkin, and Ian H Witten. 1994. Weka: A machine learning workbench. In *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on*, pages 357–361. IEEE.

Mindaugas Idzelis. 2005. Jazzy: The java open source spell checker.

Jay J Jiang and David W Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Thomas K Landauer, Danielle S McNamara, Simon Dennis, and Walter Kintsch. 2007. *Handbook of latent semantic analysis*. Psychology Press.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283.

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

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Nobal B. Niraula, Rajendra Banjade, Dan Ştefănescu, and Vasile Rus. 2013. Experiments with semantic similarity measures based on lda and lsa. In *Statistical Language and Speech Processing*, pages 188–199. Springer.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*.

Vasile Rus and Mihai Lintean. 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*, pages 157–162. Association for Computational Linguistics.

Vasile Rus, Mihai Lintean, Cristian Moldovan, William Baggett, Nobal Niraula, and Brent Morgan. 2012. The similar corpus: A resource to foster the qualitative understanding of semantic similarity of texts. In *Semantic Relations II: Enhancing Resources and Applications, The 8th Language Resources and Evaluation Conference (LREC 2012), May*, pages 23–25.

Vasile Rus, Mihai Lintean, Rajendra Banjade, Nobal Niraula, and Dan Stefanescu. 2013. Semilar: The semantic similarity toolkit. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Vasile Rus, Rajendra Banjade, and Mihai Lintean. 2014. On paraphrase identification corpora. In *Proceeding on the International Conference on Language Resources and Evaluation (LREC 2014)*.

Dan Stefanescu, Rajendra Banjade, and Vasile Rus. 2014a. Latent semantic analysis models on wikipedia and tasa.

Dan Ştefănescu, Rajendra Banjade, and Vasile Rus. 2014b. A sentence similarity method based on chunking and information content. In *Computational Linguistics and Intelligent Text Processing*, pages 442–453. Springer.

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