UNT: A Supervised Synergistic Approach to Semantic Text Similarity

Carmen Banea, Samer Hassan, Michael Mohler, Rada Mihalcea
University of North Texas
Denton, TX, USA
{CarmenBanea,SamerHassan,MichaelMohler}@my.unt.edu, rada@cs.unt.edu

Abstract

This paper presents the systems that we participated with in the Semantic Text Similarity task at SEMEVAL 2012. Based on prior research in semantic similarity and relatedness, we combine various methods in a machine learning framework. The three variations submitted during the task evaluation period ranked number 5, 9 and 14 among the 89 participating systems. Our evaluations show that corpus-based methods display a more robust behavior on the training data, yet combining a variety of methods allows a learning algorithm to achieve a superior decision than that achievable by any of the individual parts.

1 Introduction

Measures of text similarity have been used for a long time in applications in natural language processing and related areas. One of the earliest applications of text similarity is perhaps the vector-space model used in information retrieval, where the document most relevant to an input query is determined by ranking documents in a collection in reversed order of their similarity to the given query (Salton and Lesk, 1971). Text similarity has also been used for relevance feedback and text classification (Rocchio, 1971), word sense disambiguation (Lesk, 1986; Schutze, 1998), and more recently for extractive summarization (Salton et al., 1997), and methods for automatic evaluation of machine translation (Papineni et al., 2002) or text summarization (Lin and Hovy, 2003). Measures of text similarity were also found useful for the evaluation of text coherence (Lapata and Barzilay, 2005).

Earlier work on this task has primarily focused on simple lexical matching methods, which produce a similarity score based on the number of lexical units that occur in both input segments. Improvements to this simple method have considered stemming, stop-word removal, part-of-speech tagging, longest subsequence matching, as well as various weighting and normalization factors (Salton and Buckley, 1997). While successful to a certain degree, these lexical similarity methods cannot always identify the *semantic* similarity of texts. For instance, there is an obvious similarity between the text segments *I own a dog* and *I have an animal*, but most of the current text similarity metrics will fail in identifying any kind of connection between these texts.

More recently, researchers have started to consider the possibility of combining the large number of word-to-word semantic similarity measures (e.g., (Jiang and Conrath, 1997; Leacock and Chodorow, 1998; Lin, 1998; Resnik, 1995)) within a semantic similarity method that works for entire texts. The methods proposed to date in this direction mainly consist of either bipartite-graph matching strategies that aggregate word-to-word similarity into a text similarity score (Mihalcea et al., 2006; Islam and Inkpen, 2009; Hassan and Mihalcea, 2011; Mohler et al., 2011), or data-driven methods that perform component-wise additions of semantic vector representations as obtained with corpus measures such as Latent Semantic Analysis (Landauer et al., 1997), Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007), or Salient Semantic Analysis (Hassan and Mihalcea, 2011).

In this paper, we describe the system with which

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 635–642,
Montréal, Canada, June 7-8, 2012.

we participated in the SEMEVAL 2012 task on semantic text similarity (Agirre et al., 2012). The system builds upon our earlier work on corpus-based and knowledge-based methods of text semantic similarity (Mihalcea et al., 2006; Hassan and Mihalcea, 2011; Mohler et al., 2011), and combines all these previous methods into a meta-system by using machine learning. The framework provided by the task organizers also enabled us to perform an in-depth analysis of the various components used in our system, and draw conclusions concerning the role played by the different resources, features, and algorithms in building a state-of-the-art semantic text similarity system.

2 Related Work

Over the past years, the research community has focused on computing semantic relatedness using methods that are either knowledge-based or corpus-based. Knowledge-based methods derive a measure of relatedness by utilizing lexical resources and ontologies such as WordNet (Miller, 1995) to measure definitional overlap, term distance within a graphical taxonomy, or term depth in the taxonomy as a measure of specificity. We explore several of these measures in depth in Section 3.3.1. On the other side, corpus-based measures such as Latent Semantic Analysis (LSA) (Landauer et al., 1997), Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2007), Salient Semantic Analysis (SSA) (Hassan and Mihalcea, 2011), Pointwise Mutual Information (PMI) (Church and Hanks, 1990), PMI-IR (Turney, 2001), Second Order PMI (Islam and Inkpen, 2006), Hyperspace Analogues to Language (Burgess et al., 1998) and distributional similarity (Lin, 1998) employ probabilistic approaches to decode the semantics of words. They consist of unsupervised methods that utilize the contextual information and patterns observed in raw text to build semantic profiles of words. Unlike knowledge-based methods, which suffer from limited coverage, corpus-based measures are able to induce a similarity between any given two words, as long as they appear in the very large corpus used as training.

3 Semantic Textual Similarity System

The system we proposed for the SEMEVAL 2012 Semantic Textual Similarity task builds upon both knowledge- and corpus-based methods previously described in (Mihalcea et al., 2006; Hassan and Mihalcea, 2011; Mohler et al., 2011). The predictions of these independent systems, paired with additional salient features, are leveraged by a meta-system that employs machine learning. In this section, we will elaborate further on the resources we use, our features, and the components of our machine learning system. We will start by describing the task setup.

3.1 Task Setup

The training data released by the task organizers consists of three datasets showcasing two sentences per line and a manually assigned similarity score ranging from 0 (no relation) to 5 (semantically equivalent). The datasets[1] provided are taken from the Microsoft Research Paraphrase Corpus ($MSRpar$), the Microsoft Research Video Description Corpus ($MSRvid$), and the WMT2008 development dataset (Europarl section)($SMTeuroparl$); they each consist of about 750 sentence pairs with the class distribution varying with each dataset. The testing data contains additional sentences from the same collections as the training data as well as from two additional unknown sets ($OnWN$ and $SMTnews$); they range from 399 to 750 sentence pairs. The reader may refer to (Agirre et al., 2012) for additional information regarding this task.

3.2 Resources

Wikipedia[2] is a free on-line encyclopedia, representing the outcome of a continuous collaborative effort of a large number of volunteer contributors. Virtually any Internet user can create or edit a Wikipedia web page, and this "freedom of contribution" has a positive impact on both the quantity (fast-growing number of articles) and the quality (potential mistakes are quickly corrected within the collaborative environment) of this on-line resource. The basic entry in Wikipedia is an *article* which describes an entity or an event, and which, in addition to untagged

[1] http://www.cs.york.ac.uk/semeval-2012/task6/data/uploads/datasets/train-readme.txt

[2] www.wikipedia.org

content, also consists of hyperlinked text to other pages within or outside of Wikipedia. These hyperlinks are meant to guide the reader to pages that provide additional information / clarifications, so that a better understanding of the primary concept can be achieved. The structure of Wikipedia in terms of pages and hyperlinks is exploited directly by semantic similarity methods such as ESA (Gabrilovich and Markovitch, 2007), or SSA (Hassan and Mihalcea, 2011).

WordNet (Miller, 1995) is a manually crafted lexical resource that maintains semantic relationships between basic units of meaning, or *synsets*. A synset groups together senses of different words that share a very similar meaning, which act in a particular context as synonyms. Each synset is accompanied by a *gloss* or definition, and one or two examples illustrating usage in the given context. Unlike a traditional thesaurus, the structure of WordNet is able to encode additional relationships beside synonymy, such as antonymy, hypernymy, hyponymy, meronymy, entailment, etc., which various knowledge-based methods use to derive semantic similarity.

3.3 Features

Our meta-system uses several features, which can be grouped into knowledge-based, corpus-based, and bipartite graph matching, as described below. The abbreviations appearing between parentheses by each method allow for easy cross-referencing with the evaluations provided in Table 1.

3.3.1 Knowledge-based Semantic Similarity Features

Following prior work from our group (Mihalcea et al., 2006; Mohler and Mihalcea, 2009), we employ several WordNet-based similarity metrics for the task of sentence-level similarity. Briefly, for each open-class word in one of the input texts, we compute the maximum semantic similarity (using the WordNet::Similarity package (Pedersen et al., 2004)) that can be obtained by pairing it with any open-class word in the other input text. All the word-to-word similarity scores obtained in this way are summed and normalized to the length of the two input texts. We provide below a short description for each of the similarity metrics employed by this system[3].

The **shortest path** ($Path$) similarity is determined as:

$$Sim_{path} = \frac{1}{length} \quad (1)$$

where $length$ is the length of the shortest path between two concepts using node-counting (including the end nodes).

The **Leacock & Chodorow** (Leacock and Chodorow, 1998) (LCH) similarity is determined as:

$$Sim_{lch} = -\log \frac{length}{2 * D} \quad (2)$$

where $length$ is the length of the shortest path between two concepts using node-counting, and D is the maximum depth of the taxonomy.

The **Lesk** ($Lesk$) similarity of two concepts is defined as a function of the overlap between the corresponding definitions, as provided by a dictionary. It is based on an algorithm proposed by Lesk (1986) as a solution for word sense disambiguation.

The **Wu & Palmer** (Wu and Palmer, 1994) (WUP) similarity metric measures the depth of two given concepts in the WordNet taxonomy, and the depth of the least common subsumer (LCS), and combines these figures into a similarity score:

$$Sim_{wup} = \frac{2 * depth(LCS)}{depth(concept_1) + depth(concept_2)} \quad (3)$$

The measure introduced by **Resnik** (Resnik, 1995) (RES) returns the information content (IC) of the LCS of two concepts:

$$Sim_{res} = IC(LCS) \quad (4)$$

where IC is defined as:

$$IC(c) = -\log P(c) \quad (5)$$

and $P(c)$ is the probability of encountering an instance of concept c in a large corpus.

The measure introduced by **Lin** (Lin, 1998) (Lin) builds on Resnik's measure of similarity, and adds a normalization factor consisting of the information content of the two input concepts:

$$Sim_{lin} = \frac{2 * IC(LCS)}{IC(concept_1) + IC(concept_2)} \quad (6)$$

[3]We point out that the similarity metric proposed by Hirst & St. Onge was not considered due to the time constraints associated with the STS task.

We also consider the **Jiang & Conrath** (Jiang and Conrath, 1997) (JCN) measure of similarity:

$$Sim_{jnc} = \frac{1}{IC(concept_1) + IC(concept_2) - 2 * IC(LCS)} \quad (7)$$

Each of the measures listed above is used as a feature by our meta-system.

3.3.2 Corpus-based Semantic Similarity Features

While most of the corpus-based methods induce semantic profiles in a word-space, where the semantic profile of a word is expressed in terms of its co-occurrence with other words, LSA, ESA and SSA stand out as different, since they rely on a concept-space representation. In these methods, the semantic profile of a word is expressed in terms of the implicit (LSA), explicit (ESA), or salient (SSA) concepts. This departure from the sparse word-space to a denser, richer, and unambiguous concept-space resolves one of the fundamental problems in semantic relatedness, namely the vocabulary mismatch. In the experiments reported in this paper, all the corpus-based methods are trained on the English Wikipedia download from October 2008, with approximately 6 million articles, and more than 9.5 million hyperlinks.

Latent Semantic Analysis (LSA) (Landauer et al., 1997). In LSA, term-context associations are captured by means of a dimensionality reduction operated by a singular value decomposition (SVD) on the term-by-context matrix $\mathbf{T}$, where the matrix is induced from a large corpus. This reduction entails the abstraction of meaning by collapsing similar contexts and discounting noisy and irrelevant ones, hence transforming the real world term-context space into a word-latent-concept space which achieves a much deeper and concrete semantic representation of words.

Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2007). ESA uses encyclopedic knowledge in an information retrieval framework to generate a semantic interpretation of words. Since encyclopedic knowledge is typically organized into concepts (or topics), each concept is further described using definitions and examples. ESA relies on the distribution of words inside the encyclopedic descriptions. It builds semantic representations for a given word using a word-document association, where the document represents a Wikipedia article (concept). ESA is in effect a Vector Space Model (VSM) built using Wikipedia corpus, where vectors represents word-articles association.

Salient Semantic Analysis (SSA) (Hassan and Mihalcea, 2011). SSA incorporates a similar semantic abstraction and interpretation of words as ESA, yet it uses salient concepts gathered from encyclopedic knowledge, where a "concept" represents an unambiguous word or phrase with a concrete meaning, and which affords an encyclopedic definition. Saliency in this case is determined based on the word being hyperlinked (either trough manual or automatic annotations) in context, implying that they are highly relevant to the given text. SSA is an example of Generalized Vector Space Model (GVSM), where vectors represent word-concepts associations.

In order to determine the similarity of two text fragments , we employ two variations: the typical cosine similarity (cos) and a best alignment strategy ($align$), which we explain in more detail below. Both variations were paired with the LSA, ESA, and SSA systems resulting in six similarity scores that were used as features by our meta-system, namely LSA_{cos}, LSA_{align}, ESA_{cos}, ESA_{align}, SSA_{cos}, and SSA_{align}.

Best Alignment Strategy ($align$). Let T_a and T_b be two text fragments of size a and b respectively. After removing all stopwords, we first determine the number of shared terms (ω) between T_a and T_b. Second, we calculate the semantic relatedness of all possible pairings between non-shared terms in T_a and T_b. We further filter these possible combinations by creating a list φ which holds the strongest semantic pairings between the fragments' terms, such that each term can only belong to one and only one pair.

$$Sim(T_a, T_b) = \frac{(\omega + \sum_{i=1}^{|\varphi|} \varphi_i) \times (2ab)}{a + b} \quad (8)$$

where ω is the number of shared terms between the text fragments and φ_i is the similarity score for the ith pairing.

3.3.3 Bipartite Graph Matching

In an attempt to move beyond the bag-of-words paradigm described thus far, we attempt to compute

a set of dependency graph alignment scores based on previous work in automatic short-answer grading (Mohler et al., 2011). This score, computed in two stages, is used as a feature by our meta-system.

In the first stage, the system is provided with the dependency graphs for each pair of sentences[4]. For each node in one dependency graph, we compute a similarity score for each node in the other dependency graph based upon a set of lexical, semantic, and syntactic features applied to both the pair of nodes and their corresponding subgraphs (i.e. the set of nodes reachable from a given node by following directional governor-to-dependant links). The scoring function is trained on a small set of manually aligned graphs using the averaged perceptron algorithm.

We define a total of 64 features[5] to be used to train a machine learning system to compute subgraph-subgraph similarity. Of these, 32 are based upon the bag-of-words semantic similarity of the subgraphs using the metrics described in Section 3.3.1 as well as a Wikipedia-trained LSA model. The remaining 32 features are lexico-syntactic features associated with the parent nodes of the subgraphs and are described in more detail in our earlier paper.

We then calculate weights associated with these features using an averaged version of the perceptron algorithm (Freund and Schapire, 1999; Collins, 2002) trained on a set of 32 manually annotated instructor/student answer pairs selected from the short-answer grading corpus (MM2011). These pairs contain 7303 node pairs (656 matches, 6647 non-matches). Once the weights are calculated, a similarity score for each pair of nodes can be computed by taking the dot product of the feature vector with the weights.

In the second stage, the node similarity scores calculated in the previous step are used to find an optimal alignment for the pair of dependency graphs. We begin with a bipartite graph where each node in one graph is represented by a node on the left side of the bipartite graph and each node in the other graph is represented by a node on the right side. The weight associated with each edge is the score computed for each node-node pair in the previous stage. The bipartite graph is then augmented by adding dummy nodes to both sides which are allowed to match any node with a score of zero. An optimal alignment between the two graphs is then computed efficiently using the Hungarian algorithm. Note that this results in an optimal matching, not a mapping, so that an individual node is associated with at most one node in the other answer. After finding the optimal match, we produce four alignment-based scores by optionally normalizing by the number of nodes and/or weighting the node-alignments according to the idf scores of the words.[6] This results in four alignment scores listed as $graph_{none}$, $graph_{norm}$, $graph_{idf}$, $graph_{idfnorm}$.

3.3.4 Baselines

As a baseline, we also utilize several lexical bag-of-words approaches where each sentence is represented by a vector of tokens and the similarity of the two sentences can be computed by finding the cosine of the angle between their representative vectors using term frequency (tf) or term frequency multiplied by inverse document frequency ($tf.idf$)[6], or by using simple overlap between the vectors' dimensions ($overlap$).

3.4 Machine Learning

3.4.1 Algorithms

All the systems described above are used to generate a score for each training and test sample (see Section 3.1). These scores are then aggregated per sample, and used in a supervised learning framework. We decided to use a regression model, instead of classification, since the requirements for the task specify that we should provide a score in the range of 0 to 5. We could have used classification paired with bucketed ranges, yet classification does not take into consideration the underlying ordinality of the scores (i.e. a score of 4.5 is closer to either 4 or 5, but farther away from 0), which is a noticeable handicap in this scenario. We tried both linear and sup-

[4]We here use the output of the Stanford Dependency Parser in collapse/propagate mode with some modifications as described in our earlier work.

[5]With the exception of the four features based upon the Hirst & St.Onge similarity metric, these are equivalent to the features used in previous work.

[6]The document frequency scores were taken from the British National Corpus (BNC).

port vector regression[7] by performing 10 fold cross-validation on the train data, yet the latter algorithm consistently performs better, no matter what kernel was chosen. Thus we decided to use support vector regression (Smola and Schoelkopf, 1998) with a Pearson VII function-based kernel.

Due to its different learning methodology, and since it is suited for predicting continuous classes, our second system uses the M5P decision tree algorithm (Quinlan, 1992; Wang and Witten, 1997), which outperforms support vector regression on the 10 fold cross-validation performed on the SMTeuroparl train set, while providing competitive results on the other train sets (within .01 Pearson correlation).

3.4.2 Setup

We submitted three system variations, namely $IndividualRegression$, $IndividualDecTree$, and $CombinedRegression$. The first word describes the training data; for **individual**, for the *known test sets* we trained on the corresponding train sets, while for the *unknown test sets* we trained on all the train sets combined; for **combined**, for each test set we trained on all the train sets combined. The second word refers to the learning methodology, where **Regression** stands for support vector regression, and **DecTree** stands for M5P decision tree.

4 Results and Discussion

We include in Table 1 the Pearson correlations obtained by comparing the predictions of each feature to the gold standard for the three train datasets. We notice that the corpus based metrics display a consistent performance across the three train sets, when compared to the other methods, including knowledge-based. Furthermore, the best alignment strategy ($align$) for corpus based models outperforms similarity scores based on traditional cosine similarity. It is interesting to note that simple baselines such as tf, $tf.idf$ and $overlap$ offer significant correlations with all the train sets without access to additional knowledge inferred by knowledge or corpus-based methods. In the case of the bipartite graph matching, the $graph_{norm}$ variation provides the strongest correlation results across all the datasets.

System	MSRpar	MSRvid	SMTeuroparl
$Path$	0.49	0.62	0.50
LCH	0.48	0.49	0.45
$Lesk$	0.48	0.59	0.50
WUP	0.46	0.38	0.42
RES	0.47	0.55	0.48
Lin	0.49	0.54	0.48
JCN	0.49	0.63	0.51
LSA_{align}	0.44	0.57	0.61
LSA_{cos}	0.37	**0.74**	0.56
ESA_{align}	**0.52**	0.70	0.62
ESA_{cos}	0.30	0.71	0.53
SSA_{align}	0.46	0.61	**0.65**
SSA_{cos}	0.22	0.63	0.39
$graph_{none}$	0.42	0.50	0.21
$graph_{norm}$	0.48	0.43	0.59
$graph_{idf}$	0.16	0.67	0.16
$graph_{idfnorm}$	0.08	0.60	0.19
$tf.idf$	0.45	0.63	0.41
tf	0.45	0.69	0.51
$overlap$	0.44	0.69	0.27

Table 1: Correlation of individual features for the training sets with the gold standards

We include the evaluation results provided by the task organizers in Table 2. They indicate that our intuition in using a support vector regression strategy was correct. While the $IndividualRegression$ was our strongest system on the training data, the same ranking applies to the test data (including the additional two surprise datasets) as well, earning it the fifth place among the 89 participating systems, with a Pearson correlation of 0.7846.

Regarding the decision tree based learning ($IndividualDecTree$), despite its more robust behavior on the train sets, it achieved slightly lower outcome on the test data, at 0.7677 correlation. We believe this happened because decision trees have a tendency to overfit training data, as they generate a rigid structure which is unforgiving to minor deviations in the test data. Nonetheless, this second variation still ranks in the top 10% of the submitted systems.

As an alternative approach to handle unknown test data (e.g. different distributions, genres), we opted

[7] Implementations provided through the Weka framework (Hall et al., 2009).

Run	ALL	Rank	Mean	RankMean	MSRpar	MSRvid	SMTeuroparl	OnWN	SMTnews
$IndividualRegression$	**0.7846**	**5**	0.6162	13	0.5353	0.8750	0.4203	0.6715	0.4033
$IndividualDecTree$	**0.7677**	**9**	0.5947	25	0.5693	0.8688	0.4203	0.6491	0.2256
$CombinedRegression$	**0.7418**	**14**	0.6159	14	0.5032	0.8695	0.4797	0.6715	0.4033

Table 2: Evaluation results and ranking published by the task organizers

to also include the $CombinedRegression$ strategy as our third variation. This seems to have been fruitful for $MSRvid$, $SMTeuroparl$, and the two surprise datasets ($ONWn$ and $SMTnews$). In the case of $SMTeuroparl$, this expanded training set achieves a better performance than learning from the corresponding training set alone, gaining an improvement of 0.0776 correlation points. Unfortunately, the variation has some losses, particularly for the $MSRpar$ dataset (0.0321), yet it is able to consistently model and handle a wider variety of text types.

5 Conclusion

This paper describes the three system variations our team participated with in the Semantic Text Similarity task in SEMEVAL 2012. Our focus has been to produce a synergistic approach, striving to achieve a superior result than attainable by each system individually. We have considered a variety of methods for inferring semantic similarity, including knowledge and corpus-based methods. These were leveraged in a machine-learning framework, where our preferred learning algorithm is support vector regression, due to its ability to deal with continuous classes and to dampen the effect of noisy features, while augmenting more robust ones. While it is always preferable to use similar test and train sets, when information regarding the test dataset is unavailable, we show that a robust performance can be achieved by combining all train data from different sources into a single set and allowing a machine learner to make predictions. Overall, it was interesting to note that corpus-based methods maintain strong results on all train datasets in comparison to knowledge-based methods. Our three systems ranked number 5, 9 and 14 among the 89 systems participating in the task.

Acknowledgments

This material is based in part upon work supported by the National Science Foundation CAREER award #0747340 and IIS award #1018613. 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 National Science Foundation.

References

E. Agirre, D. Cer, M. Diab, and A. Gonzalez. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), in conjunction with the First Joint Conference on Lexical and Computational Semantics (*SEM 2012)*.

C. Burgess, K. Livesay, and K. Lund. 1998. Explorations in context space: words, sentences, discourse. *Discourse Processes*, 25(2):211–257.

K. Church and P. Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

M. Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP-02)*, Philadelphia, PA, July.

Y. Freund and R. Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning*, 37:277–296.

E. Gabrilovich and S. Markovitch. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1606–1611, Hyderabad, India.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1).

S. Hassan and R. Mihalcea. 2011. Measuring semantic relatedness using salient encyclopedic concepts. *Artificial Intelligence, Special Issue*, xx(xx).

A. Islam and D. Inkpen. 2006. Second order co-occurrence PMI for determining the semantic similarity of words. In *Proceedings of the Fifth Conference on Language Resources and Evaluation*, volume 2, Genoa, Italy, July.

A. Islam and D. Inkpen. 2009. Semantic Similarity of Short Texts. In Nicolas Nicolov, Galia Angelova, and Ruslan Mitkov, editors, *Recent Advances in Natural Language Processing V*, volume 309 of *Current Issues in Linguistic Theory*, pages 227–236. John Benjamins, Amsterdam & Philadelphia.

J. J. Jiang and D. W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *International Conference Research on Computational Linguistics (ROCLING X)*, pages 9008+, September.

T. K. Landauer, D. Laham, B. Rehder, and M. E. Schreiner. 1997. How well can passage meaning be derived without using word order? a comparison of latent semantic analysis and humans.

M. Lapata and R. Barzilay. 2005. Automatic evaluation of text coherence: Models and representations. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, Edinburgh.

C. Leacock and M. Chodorow, 1998. *Combining local context and WordNet similarity for word sense identification*, pages 305–332.

M. Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26, New York, NY, USA. ACM.

C. Y. Lin and E. H. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of Human Language Technology Conference (HLT-NAACL 2003)*, Edmonton, Canada, May.

D. Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, Madison, Wisconsin.

R. Mihalcea, C. Corley, and C. Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the American Association for Artificial Intelligence (AAAI 2006)*, pages 775–780, Boston, MA, US.

G. A. Miller. 1995. WordNet: a Lexical database for english. *Communications of the Association for Computing Machinery*, 38(11):39–41.

M. Mohler and R. Mihalcea. 2009. Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the European Association for Computational Linguistics (EACL 2009)*, Athens, Greece.

M. Mohler, R. Bunescu, and R. Mihalcea. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the Association for Computational Linguistics – Human Language Technologies (ACL-HLT 2011)*, Portland, Oregon, USA.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA.

T. Pedersen, S. Patwardhan, and J. Michelizzi. 2004. WordNet:: Similarity-Measuring the Relatedness of Concepts. *Proceedings of the National Conference on Artificial Intelligence*, pages 1024–1025.

R. J. Quinlan. 1992. Learning with continuous classes. In *5th Australian Joint Conference on Artificial Intelligence*, pages 343–348, Singapore. World Scientific.

P. Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *In Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453.

J. Rocchio, 1971. *Relevance feedback in information retrieval*. Prentice Hall, Ing. Englewood Cliffs, New Jersey.

G. Salton and C. Buckley. 1997. Term weighting approaches in automatic text retrieval. In *Readings in Information Retrieval*. Morgan Kaufmann Publishers, San Francisco, CA.

G. Salton and M.E. Lesk, 1971. *The SMART Retrieval System: Experiments in Automatic Document Processing*, chapter Computer evaluation of indexing and text processing. Prentice Hall, Ing. Englewood Cliffs, New Jersey.

G. Salton, A. Singhal, M. Mitra, and C. Buckley. 1997. Automatic text structuring and summarization. *Information Processing and Management*, 2(32).

H. Schutze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.

A. J. Smola and B. Schoelkopf. 1998. A tutorial on support vector regression. NeuroCOLT2 Technical Report NC2-TR-1998-030.

P. D. Turney. 2001. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning*, pages 491–502, Freiburg, Germany.

Y. Wang and I. H. Witten. 1997. Induction of model trees for predicting continuous classes. In *Poster papers of the 9th European Conference on Machine Learning*. Springer.

Z. Wu and M. Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133—-138, Las Cruces, New Mexico.