Distributional Semantics Beyond Words: Supervised Learning of Analogy and Paraphrase

Peter D. Turney
National Research Council Canada
Information and Communications Technologies
Ottawa, Ontario, Canada, K1A 0R6
peter.turney@nrc-cnrc.gc.ca

Abstract

There have been several efforts to extend distributional semantics beyond individual words, to measure the similarity of word pairs, phrases, and sentences (briefly, *tuples*; ordered sets of words, contiguous or noncontiguous). One way to extend beyond words is to compare two tuples using a function that combines pairwise similarities between the component words in the tuples. A strength of this approach is that it works with both relational similarity (analogy) and compositional similarity (paraphrase). However, past work required hand-coding the combination function for different tasks. The main contribution of this paper is that combination functions are generated by supervised learning. We achieve state-of-the-art results in measuring relational similarity between word pairs (SAT analogies and SemEval 2012 Task 2) and measuring compositional similarity between noun-modifier phrases and unigrams (multiple-choice paraphrase questions).

1 Introduction

Harris (1954) and Firth (1957) hypothesized that words that appear in similar contexts tend to have similar meanings. This hypothesis is the foundation for distributional semantics, in which words are represented by context vectors. The similarity of two words is calculated by comparing the two corresponding context vectors (Lund et al., 1995; Landauer and Dumais, 1997; Turney and Pantel, 2010).

Distributional semantics is highly effective for measuring the semantic similarity between individual words. On a set of eighty multiple-choice synonym questions from the test of English as a foreign language (TOEFL), a distributional approach recently achieved 100% accuracy (Bullinaria and Levy, 2012). However, it has been difficult to extend distributional semantics beyond individual words, to word pairs, phrases, and sentences.

Moving beyond individual words, there are various types of semantic similarity to consider. Here we focus on paraphrase and analogy. Paraphrase is similarity in the meaning of two pieces of text (Androutsopoulos and Malakasiotis, 2010). Analogy is similarity in the semantic relations of two sets of words (Turney, 2008a).

It is common to study paraphrase at the sentence level (Androutsopoulos and Malakasiotis, 2010), but we prefer to concentrate on the simplest type of paraphrase, where a bigram paraphrases a unigram. For example, *dog house* is a paraphrase of *kennel*. In our experiments, we concentrate on noun-modifier bigrams and noun unigrams.

Analogies map terms in one domain to terms in another domain (Gentner, 1983). The familiar analogy between the solar system and the Rutherford-Bohr atomic model involves several terms from the domain of the solar system and the domain of the atomic model (Turney, 2008a).

The simplest type of analogy is proportional analogy, which involves two pairs of words (Turney, 2006b). For example, the pair ⟨*cook, raw*⟩ is analogous to the pair ⟨*decorate, plain*⟩. If we *cook* a thing, it is no longer *raw*; if we *decorate* a thing, it

Transactions of the Association for Computational Linguistics, 1 (2013) 353–366. Action Editor: Patrick Pantel.
Submitted 5/2013; Revised 7/2013; Published 10/2013.

is no longer *plain*. The semantic relations between *cook* and *raw* are similar to the semantic relations between *decorate* and *plain*. In the following experiments, we focus on proportional analogies.

Erk (2013) distinguished four approaches to extend distributional semantics beyond words: In the first, a single vector space representation for a phrase or sentence is computed from the representations of the individual words (Mitchell and Lapata, 2010; Baroni and Zamparelli, 2010). In the second, two phrases or sentences are compared by combining multiple pairwise similarity values (Socher et al., 2011; Turney, 2012). Third, weighted inference rules integrate distributional similarity and formal logic (Garrette et al., 2011). Fourth, a single space integrates formal logic and vectors (Clarke, 2012).

Taking the second approach, Turney (2012) introduced a dual-space model, with one space for measuring domain similarity (similarity of topic or field) and another for function similarity (similarity of role or usage). Similarities beyond individual words are calculated by functions that combine domain and function similarities of component words.

The dual-space model has been applied to measuring compositional similarity (paraphrase recognition) and relational similarity (analogy recognition). In experiments that tested for sensitivity to word order, the dual-space model performed significantly better than competing approaches (Turney, 2012).

A limitation of past work with the dual-space model is that the combination functions were hand-coded. Our main contribution is to show how hand-coding can be eliminated with supervised learning. For ease of reference, we will call our approach *SuperSim* (supervised similarity). With no modification of SuperSim for the specific task (relational similarity or compositional similarity), we achieve better results than previous hand-coded models.

Compositional similarity (paraphrase) compares two contiguous phrases or sentences (n-grams), whereas relational similarity (analogy) does not require contiguity. We use *tuple* to refer to both contiguous and noncontiguous word sequences.

We approach analogy as a problem of supervised tuple classification. To measure the relational similarity between two word pairs, we train SuperSim with quadruples that are labeled as positive and negative examples of analogies. For example, the proportional analogy $\langle$*cook, raw, decorate, plain*$\rangle$ is labeled as a positive example.

A quadruple is represented by a feature vector, composed of domain and function similarities from the dual-space model and other features based on corpus frequencies. SuperSim uses a support vector machine (Platt, 1998) to learn the probability that a quadruple $\langle a, b, c, d\rangle$ consists of a word pair $\langle a, b\rangle$ and an analogous word pair $\langle c, d\rangle$. The probability can be interpreted as the degree of relational similarity between the two given word pairs.

We also approach paraphrase as supervised tuple classification. To measure the compositional similarity beween an m-gram and an n-gram, we train the learning algorithm with $(m + n)$-tuples that are positive and negative examples of paraphrases.

SuperSim learns to estimate the probability that a triple $\langle a, b, c\rangle$ consists of a compositional bigram ab and a synonymous unigram c. For instance, the phrase *fish tank* is synonymous with *aquarium*; that is, *fish tank* and *aquarium* have high compositional similarity. The triple $\langle$*fish, tank, aquarium*$\rangle$ is represented using the same features that we used for analogy. The probability of the triple can be interpreted as the degree of compositional similarity between the given bigram and unigram.

We review related work in Section 2. The general feature space for learning relations and compositions is presented in Section 3. The experiments with relational similarity are described in Section 4, and Section 5 reports the results with compositional similarity. Section 6 discusses the implications of the results. We consider future work in Section 7 and conclude in Section 8.

2 Related Work

In SemEval 2012, Task 2 was concerned with measuring the degree of relational similarity between two word pairs (Jurgens et al., 2012) and Task 6 (Agirre et al., 2012) examined the degree of semantic equivalence between two sentences. These two areas of research have been mostly independent, although Socher et al. (2012) and Turney (2012) present unified perspectives on the two tasks. We first discuss some work on relational similarity, then some work on compositional similarity, and lastly work that unifies the two types of similarity.

2.1 Relational Similarity

LRA (latent relational analysis) measures relational similarity with a pair–pattern matrix (Turney, 2006b). Rows in the matrix correspond to word pairs (a, b) and columns correspond to patterns that connect the pairs ("a for the b") in a large corpus. This is a *holistic* (noncompositional) approach to distributional similarity, since the word pairs are opaque wholes; the component words have no separate representations. A compositional approach to analogy has a representation for each word, and a word pair is represented by composing the representations for each member of the pair. Given a vocabulary of N words, a compositional approach requires N representations to handle all possible word pairs, but a holistic approach requires N^2 representations. Holistic approaches do not scale up (Turney, 2012). LRA required nine days to run.

Bollegala et al. (2008) answered the SAT analogy questions with a support vector machine trained on quadruples (proportional analogies), as we do here. However, their feature vectors are holistic, and hence there are scaling problems.

Herdağdelen and Baroni (2009) used a support vector machine to learn relational similarity. Their feature vectors contained a combination of holistic and compositional features.

Measuring relational similarity is closely connected to classifying word pairs according to their semantic relations (Turney and Littman, 2005). Semantic relation classification was the focus of SemEval 2007 Task 4 (Girju et al., 2007) and SemEval 2010 Task 8 (Hendrickx et al., 2010).

2.2 Compositional Similarity

To extend distributional semantics beyond words, many researchers take the first approach described by Erk (2013), in which a single vector space is used for individual words, phrases, and sentences (Landauer and Dumais, 1997; Mitchell and Lapata, 2008; Mitchell and Lapata, 2010). In this approach, given the words a and b with context vectors $\mathbf{a}$ and $\mathbf{b}$, we construct a vector for the bigram ab by applying vector operations to $\mathbf{a}$ and $\mathbf{b}$.

Mitchell and Lapata (2010) experiment with many different vector operations and find that element-wise multiplication performs well. The bigram ab is represented by $\mathbf{c} = \mathbf{a} \odot \mathbf{b}$, where $c_i = a_i \cdot b_i$. However, element-wise multiplication is commutative, so the bigrams ab and ba map to the same vector $\mathbf{c}$. In experiments that test for order sensitivity, element-wise multiplication performs poorly (Turney, 2012).

We can treat the bigram ab as a unit, as if it were a single word, and construct a context vector for ab from occurrences of ab in a large corpus. This holistic approach to representing bigrams performs well when a limited set of bigrams is specified in advance (before building the word–context matrix), but it does not scale up, because there are too many possible bigrams (Turney, 2012).

Although the holistic approach does not scale up, we can generate a few holistic bigram vectors and use them to train a supervised regression model (Guevara, 2010; Baroni and Zamparelli, 2010). Given a new bigram cd, not observed in the corpus, the regression model can predict a holistic vector for cd, if c and d have been observed separately. We show in Section 5 that this idea can be adapted to train SuperSim without manually labeled data.

Socher et al. (2011) take the second approach described by Erk (2013), in which two sentences are compared by combining multiple pairwise similarity values. They construct a variable-sized similarity matrix $\mathbf{X}$, in which the element x_{ij} is the similarity between the i-th phrase of one sentence and the j-th phrase of the other. Since supervised learning is simpler with fixed-sized feature vectors, the variable-sized similarity matrix is then reduced to a smaller fixed-sized matrix, to allow comparison of pairs of sentences of varying lengths.

2.3 Unified Perspectives on Similarity

Socher et al. (2012) represent words and phrases with a pair, consisting of a vector and a matrix. The vector captures the meaning of the word or phrase and the matrix captures how a word or phrase modifies the meaning of another word or phrase when they are combined. They apply this matrix–vector representation to both compositions and relations.

Turney (2012) represents words with two vectors, a vector from domain space and a vector from function space. The domain vector captures the topic or field of the word and the function vector captures the

functional role of the word. This dual-space model is applied to both compositions and relations.

Here we extend the dual-space model of Turney (2012) in two ways: Hand-coding is replaced with supervised learning and two new sets of features augment domain and function space. Moving to supervised learning instead of hand-coding makes it easier to introduce new features.

In the dual-space model, parameterized similarity measures provided the input values for hand-crafted functions. Each task required a different set of hand-crafted functions. The parameters of the similarity measures were tuned using a customized grid search algorithm. The grid search algorithm was not suitable for integration with a supervised learning algorithm. The insight behind SuperSim is that, given appropriate features, a supervised learning algorithm can replace the grid search algorithm and the hand-crafted functions.

3 Features for Tuple Classification

We represent a tuple with four types of features, all based on frequencies in a large corpus. The first type of feature is the logarithm of the frequency of a word. The second type is the positive pointwise mutual information (PPMI) between two words (Church and Hanks, 1989; Bullinaria and Levy, 2007). Third and fourth are the similarities of two words in domain and function space (Turney, 2012).

In the following experiments, we use the PPMI matrix from Turney et al. (2011) and the domain and function matrices from Turney (2012).[1] The three matrices and the word frequency data are based on the same corpus, a collection of web pages gathered from university web sites, containing 5×10^{10} words.[2] All three matrices are word–context matrices, in which the rows correspond to terms (words and phrases) in WordNet.[3] The columns correspond to the contexts in which the terms appear; each matrix involves a different kind of context.

[1]The three matrices and the word frequency data are available on request from the author. The matrix files range from two to five gigabytes when packaged and compressed for distribution.

[2]The corpus was collected by Charles Clarke at the University of Waterloo. It is about 280 gigabytes of plain text.

[3]See `http://wordnet.princeton.edu/` for information about WordNet.

Let $\langle x_1, x_2, \ldots, x_n \rangle$ be an n-tuple of words. The number of features we use to represent this tuple increases as a function of n.

The first set of features consists of log frequency values for each word x_i in the n-tuple. Let freq(x_i) be the frequency of x_i in the corpus. We define LF(x_i) as log(freq(x_i)+1). If x_i is not in the corpus, freq(x_i) is zero, and thus LF(x_i) is also zero. There are n log frequency features, one LF(x_i) feature for each word in the n-tuple.

The second set of features consists of positive pointwise mutual information values for each pair of words in the n-tuple. We use the raw PPMI matrix from Turney et al. (2011). Although they computed the singular value decomposition (SVD) to project the row vectors into a lower-dimensional space, we need the original high-dimensional columns for our features. The raw PPMI matrix has 114,501 rows and 139,246 columns with a density of 1.2%. For each term in WordNet, there is a corresponding row in the raw PPMI matrix. For each unigram in WordNet, there are two corresponding columns in the raw PPMI matrix, one marked *left* and the other *right*.

Suppose x_i corresponds to the i-th row of the PPMI matrix and x_j corresponds the j-th column, marked *left*. The value in the i-th row and j-th column of the PPMI matrix, PPMI(x_i, x_j, *left*), is the positive pointwise mutual information of x_i and x_j co-occurring in the corpus, where x_j is the first word to the left of x_i, ignoring any intervening stop words (that is, ignoring any words that are not in WordNet). If x_i (or x_j) has no corresponding row (or column) in the matrix, then the PPMI value is set to zero.

Turney et al. (2011) estimated PPMI(x_i, x_j, *left*) by sampling the corpus for phrases containing x_i and then looking for x_j to the left of x_i in the sampled phrases (and likewise for *right*). Due to this sampling process, PPMI(x_i, x_j, *left*) does not necessarily equal PPMI(x_j, x_i, *right*). For example, suppose x_i is a rare word and x_j is a common word. With PPMI(x_i, x_j, *left*), when we sample phrases containing x_i, we are relatively likely to find x_j in some of these phrases. With PPMI(x_j, x_i, *right*), when we sample phrases containing x_j, we are less likely to find any phrases containing x_i. Although, in theory, PPMI(x_i, x_j, *left*) should equal PPMI(x_j, x_i, *right*), they are likely to be unequal given a limited sample.

From the n-tuple, we select all of the $n(n-1)$ pairs, $\langle x_i, x_j \rangle$, such that $i \neq j$. We then generate two features for each pair, PPMI(x_i, x_j, *left*) and PPMI(x_i, x_j, *right*). Thus there are $2n(n-1)$ PPMI values in the second set of features.

The third set of features consists of domain space similarity values for each pair of words in the n-tuple. Domain space was designed to capture the topic of a word. Turney (2012) first constructed a frequency matrix, in which the rows correspond to terms in WordNet and the columns correspond to nearby nouns. Given a term x_i, the corpus was sampled for phrases containing x_i and the phrases were processed with a part-of-speech tagger, to identify nouns. If the noun x_j was the closest noun to the left or right of x_i, then the frequency count for the i-th row and j-th column was incremented. The hypothesis was that the nouns near a term characterize the topics associated with the term.

The word–context frequency matrix for domain space has 114,297 rows (terms) and 50,000 columns (noun contexts, topics), with a density of 2.6%. The frequency matrix was converted to a PPMI matrix and then smoothed with SVD. The SVD yields three matrices, $\mathbf{U}$, $\mathbf{\Sigma}$, and $\mathbf{V}$.

A term in domain space is represented by a row vector in $\mathbf{U}_k\mathbf{\Sigma}_k^p$. The parameter k specifies the number of singular values in the truncated singular value decomposition; that is, k is the number of latent factors in the low-dimensional representation of the term (Landauer and Dumais, 1997). We generate $\mathbf{U}_k$ and $\mathbf{\Sigma}_k$ by deleting the columns in $\mathbf{U}$ and $\mathbf{\Sigma}$ corresponding to the smallest singular values. The parameter p raises the singular values in $\mathbf{\Sigma}_k$ to the power p (Caron, 2001). As p goes from one to zero, factors with smaller singular values are given more weight. This has the effect of making the similarity measure more discriminating (Turney, 2012).

The similarity of two words in domain space, Dom(x_i, x_j, k, p), is computed by extracting the row vectors in $\mathbf{U}_k\mathbf{\Sigma}_k^p$ that correspond to the words x_i and x_j, and then calculating their cosine. Optimal performance requires tuning the parameters k and p for the task (Bullinaria and Levy, 2012; Turney, 2012). In the following experiments, we avoid directly tuning k and p by generating features with a variety of values for k and p, allowing the supervised learning algorithm to decide which features to use.

Feature set	Size of set
LF(x_i)	n
PPMI(x_i, x_j, *handedness*)	$2n(n-1)$
Dom(x_i, x_j, k, p)	$\frac{1}{2}n(n-1)n_k n_p$
Fun(x_i, x_j, k, p)	$\frac{1}{2}n(n-1)n_k n_p$

Table 1: The four sets of features and their sizes.

From the n-tuple, we select all $\frac{1}{2}n(n-1)$ pairs, $\langle x_i, x_j \rangle$, such that $i < j$. For each pair, we generate domain similarity features, Dom(x_i, x_j, k, p), where k varies from 100 to 1000 in steps of 100 and p varies from 0 to 1 in steps of 0.1. The number of k values, n_k, is 10 and the number of p values, n_p, is 11; therefore there are 110 features, $n_k n_p$, for each pair, $\langle x_i, x_j \rangle$. Thus there are $\frac{1}{2}n(n-1)n_k n_p$ domain space similarity values in the third set of features.

The fourth set of features consists of function space similarity values for each pair of words in the n-tuple. Function space was designed to capture the functional role of a word. It is similar to domain space, except the context is based on verbal patterns, instead of nearby nouns. The hypothesis was that the functional role of a word is characterized by the patterns that relate the word to nearby verbs.

The word–context frequency matrix for function space has 114,101 rows (terms) and 50,000 columns (verb pattern contexts, functional roles), with a density of 1.2%. The frequency matrix was converted to a PPMI matrix and smoothed with SVD.

From the n-tuple, we select all $\frac{1}{2}n(n-1)$ pairs, $\langle x_i, x_j \rangle$, such that $i < j$. For each pair, we generate function similarity features, Fun(x_i, x_j, k, p), where k and p vary as they did with domain space. Thus there are $\frac{1}{2}n(n-1)n_k n_p$ function space similarity values in the fourth set of features.

Table 1 summarizes the four sets of features and the size of each set as a function of n, the number of words in the given tuple. The values of n_k and n_p (10 and 11) are considered to be constants. Table 2 shows the number of elements in the feature vector, as n varies from 1 to 6. The total number of features is $O(n^2)$. We believe that this is acceptable growth and will scale up to comparing sentence pairs.

The four sets of features have a hierarchical relationship. The log frequency features are based on counting isolated occurrences of each word in the

n-tuple	LF	PPMI	Dom	Fun	Total
1	1	0	0	0	1
2	2	4	110	110	226
3	3	12	330	330	675
4	4	24	660	660	1348
5	5	40	1100	1100	2245
6	6	60	1650	1650	3366

Table 2: Number of features for various tuple sizes.

Stem:		word:language
Choices:	(1)	paint:portrait
	(2)	poetry:rhythm
	(3)	note:music
	(4)	tale:story
	(5)	week:year
Solution:	(3)	note:music

Table 3: A five-choice SAT analogy question.

corpus. The PPMI features are based on direct co-occurrences of two words; that is, PPMI is only greater than zero if the two words actually occur together in the corpus. Domain and function space capture indirect or higher-order co-occurrence, due to the truncated SVD (Lemaire and Denhière, 2006); that is, the values of $\text{Dom}(x_i, x_j, k, p)$ and $\text{Fun}(x_i, x_j, k, p)$ can be high even when x_i and x_j do not actually co-occur in the corpus. We conjecture that there are yet higher orders in this hierarchy that would provide improved similarity measures.

SuperSim learns to classify tuples by representing them with these features. SuperSim uses the sequential minimal optimization (SMO) support vector machine (SVM) as implemented in Weka (Platt, 1998; Witten et al., 2011).[4] The kernel is a normalized third-order polynomial. Weka provides probability estimates for the classes by fitting the outputs of the SVM with logistic regression models.

4 Relational Similarity

This section presents experiments with learning relational similarity using SuperSim. The training datasets consist of quadruples that are labeled as positive and negative examples of analogies. Table 2 shows that the feature vectors have 1,348 elements.

We experiment with three datasets, a collection of 374 five-choice questions from the SAT college entrance exam (Turney et al., 2003), a modified ten-choice variation of the SAT questions (Turney, 2012), and the relational similarity dataset from SemEval 2012 Task 2 (Jurgens et al., 2012).[5]

[4]Weka is available at `http://www.cs.waikato.ac.nz/ml/weka/`.

[5]The SAT questions are available on request from the author. The SemEval 2012 Task 2 dataset is available at `https://sites.google.com/site/semeval2012task2/`.

4.1 Five-choice SAT Questions

Table 3 is an example of a question from the 374 five-choice SAT questions. Each five-choice question yields five labeled quadruples, by combining the stem with each choice. The quadruple $\langle$*word, language, note, music*$\rangle$ is labeled *positive* and the other four quadruples are labeled *negative*.

Since learning works better with balanced training data (Japkowicz and Stephen, 2002), we use the symmetries of proportional analogies to add more positive examples (Lepage and Shin-ichi, 1996). For each positive quadruple, $\langle a, b, c, d\rangle$, we add three more positive quadruples, $\langle b, a, d, c\rangle$, $\langle c, d, a, b\rangle$, and $\langle d, c, b, a\rangle$. Thus each five-choice question provides four positive and four negative quadruples.

We use ten-fold cross-validation to apply SuperSim to the SAT questions. The folds are constructed so that the eight quadruples from each SAT question are kept together in the same fold. To answer a question in the testing fold, the learned model assigns a probability to each of the five choices and guesses the choice with the highest probability. SuperSim achieves a score of 54.8% correct (205 out of 374).

Table 4 gives the rank of SuperSim in the list of the top ten results with the SAT analogy questions.[6] The scores ranging from 51.1% to 57.0% are not significantly different from SuperSim's score of 54.8%, according to Fisher's exact test at the 95% confidence level. However, SuperSim answers the SAT questions in a few minutes, whereas LRA requires nine days, and SuperSim learns its models automatically, unlike the hand-coding of Turney (2012).

[6]See the *State of the Art* page on the ACL Wiki at `http://aclweb.org/aclwiki`.

Algorithm	Reference	Correct
Know-Best	Veale (2004)	43.0
k-means	Biçici & Yuret (2006)	44.0
BagPack	Herdağdelen & Baroni (2009)	44.1
VSM	Turney & Littman (2005)	47.1
Dual-Space	Turney (2012)	51.1
BMI	Bollegala et al. (2009)	51.1
PairClass	Turney (2008b)	52.1
PERT	Turney (2006a)	53.5
SuperSim	—	54.8
LRA	Turney (2006b)	56.1
Human	Average college applicant	57.0

Table 4: The top ten results on five-choice SAT questions.

	Features				
Algorithm	LF	PPMI	Dom	Fun	Correct
Dual-Space	0	0	1	1	47.9
SuperSim	1	1	1	1	52.7
SuperSim	0	1	1	1	52.7
SuperSim	1	0	1	1	52.7
SuperSim	1	1	0	1	45.7
SuperSim	1	1	1	0	41.7
SuperSim	1	0	0	0	5.6
SuperSim	0	1	0	0	32.4
SuperSim	0	0	1	0	39.6
SuperSim	0	0	0	1	39.3

Table 5: Feature ablation with ten-choice SAT questions.

4.2 Ten-choice SAT Questions

In addition to symmetries, proportional analogies have asymmetries. In general, if the quadruple $\langle a, b, c, d\rangle$ is positive, $\langle a, d, c, b\rangle$ is negative. For example, $\langle$*word, language, note, music*$\rangle$ is a good analogy, but $\langle$*word, music, note, language*$\rangle$ is not. *Words* are the basic units of *language* and *notes* are the basic units of *music*, but *words* are not necessary for *music* and *notes* are not necessary for *language*.

Turney (2012) used this asymmetry to convert the 374 five-choice SAT questions into 374 ten-choice SAT questions. Each choice $\langle c, d\rangle$ was expanded with the stem $\langle a, b\rangle$, resulting in the quadruple $\langle a, b, c, d\rangle$, and then the order was shuffled to $\langle a, d, c, b\rangle$, so that each choice pair in a five-choice question generated two choice quadruples in a ten-choice question. Nine of the quadruples are negative examples and the quadruple consisting of the stem pair followed by the solution pair is the only positive example. The purpose of the ten-choice questions is to test the ability of measures of relational similarity to avoid the asymmetric distractors.

We use the ten-choice questions to compare the hand-coded dual-space approach (Turney, 2012) with SuperSim. We also use these questions to perform an ablation study of the four sets of features in SuperSim. As with the five-choice questions, we use the symmetries of proportional analogies to add three more positive examples, so the training dataset has nine negative examples and four positive examples per question. We apply ten-fold cross-validation to the 374 ten-choice questions.

On the ten-choice questions, SuperSim's score is 52.7% (Table 5), compared to 54.8% on the five-choice questions (Table 4), a drop of 2.1%. The hand-coded dual-space model scores 47.9% (Table 5), compared to 51.1% on the five-choice questions (Table 4), a drop of 3.2%. The difference between SuperSim (52.7%) and the hand-coded dual-space model (47.9%) is not significant according to Fisher's exact test at the 95% confidence level. The advantage of SuperSim is that it does not need hand-coding. The results show that SuperSim can avoid the asymmetric distractors.

Table 5 shows the impact of different subsets of features on the percentage of correct answers to the ten-choice SAT questions. Included features are marked 1 and ablated features are marked 0. The results show that the log frequency (LF) and PPMI features are not helpful (but also not harmful) for relational similarity. We also see that domain space and function space are both needed for good results.

4.3 SemEval 2012 Task 2

The SemEval 2012 Task 2 dataset is based on the semantic relation classification scheme of Bejar et al. (1991), consisting of ten high-level categories of relations and seventy-nine subcategories, with paradigmatic examples of each subcategory. For instance, the subcategory *taxonomic* in the category *class inclusion* has three paradigmatic examples, *flower:tulip*, *emotion:rage*, and *poem:sonnet*.

Jurgens et al. (2012) used Amazon's Mechanical Turk to create the SemEval 2012 Task 2 dataset in two phases. In the first phase, Turkers expanded the paradigmatic examples for each subcategory to an

Algorithm	Reference	Spearman
BUAP	Tovar et al. (2012)	0.014
Duluth-V2	Pedersen (2012)	0.038
Duluth-V1	Pedersen (2012)	0.039
Duluth-V0	Pedersen (2012)	0.050
UTD-SVM	Rink & Harabagiu (2012)	0.116
UTD-NB	Rink & Harabagiu (2012)	0.229
RNN-1600	Mikolov et al. (2013)	0.275
UTD-LDA	Rink & Harabagiu (2013)	0.334
Com	Zhila et al. (2013)	0.353
SuperSim	—	0.408

Table 6: Spearman correlations for SemEval 2012 Task 2.

Stem:		fantasy world
Choices:	(1)	fairyland
	(2)	fantasy
	(3)	world
	(4)	phantasy
	(5)	universe
	(6)	ranter
	(7)	souring
Solution:	(1)	fairyland

Table 7: A noun-modifier question based on WordNet.

average of forty-one word pairs per subcategory, a total of 3,218 pairs. In the second phase, each word pair from the first phase was assigned a prototypicality score, indicating its similarity to the paradigmatic examples. The challenge of SemEval 2012 Task 2 was to guess the prototypicality scores.

SuperSim was trained on the five-choice SAT questions and evaluated on the SemEval 2012 Task 2 test dataset. For a given a word pair, we created quadruples, combining the word pair with each of the paradigmatic examples for its subcategory. We then used SuperSim to compute the probabilities for each quadruple. Our guess for the prototypicality score of the given word pair was the average of the probabilities. Spearman's rank correlation coefficient between the Turkers' prototypicality scores and SuperSim's scores was 0.408, averaged over the sixty-nine subcategories in the testing set. SuperSim has the highest Spearman correlation achieved to date on SemEval 2012 Task 2 (see Table 6).

5 Compositional Similarity

This section presents experiments using SuperSim to learn compositional similarity. The datasets consist of triples, $\langle a, b, c \rangle$, such that ab is a noun-modifier bigram and c is a noun unigram. The triples are labeled as positive and negative examples of paraphrases. Table 2 shows that the feature vectors have 675 elements. We experiment with two datasets, seven-choice and fourteen-choice noun-modifier questions (Turney, 2012).[7]

[7]The seven-choice dataset is available at `http://jair.org/papers/paper3640.html`. The fourteen-choice dataset can be generated from the seven-choice dataset.

5.1 Noun-Modifier Questions

The first dataset is a seven-choice noun-modifier question dataset, constructed from WordNet (Turney, 2012). The dataset contains 680 questions for training and 1,500 for testing, a total of 2,180 questions. Table 7 shows one of the questions.

The stem is a bigram and the choices are unigrams. The bigram is composed of a head noun (*world*), modified by an adjective or noun (*fantasy*). The solution is the unigram (*fairyland*) that belongs to the same WordNet synset as the stem.

The distractors are designed to be difficult for current approaches to composition. For example, if *fantasy world* is represented by element-wise multiplication of the context vectors for *fantasy* and *world* (Mitchell and Lapata, 2010), the most likely guess is *fantasy* or *world*, not *fairyland* (Turney, 2012).

Each seven-choice question yields seven labeled triples, by combining the stem with each choice. The triple $\langle$*fantasy, world, fairyland*$\rangle$ is labeled *positive* and the other six triples are labeled *negative*.

In general, if $\langle a, b, c \rangle$ is a positive example, then $\langle b, a, c \rangle$ is negative. For example, *world fantasy* is not a paraphrase of *fairyland*. The second dataset is constructed by applying this shuffling transformation to convert the 2,180 seven-choice questions into 2,180 fourteen-choice questions (Turney, 2012). The second dataset is designed to be difficult for approaches that are not sensitive to word order.

Table 8 shows the percentage of the testing questions that are answered correctly for the two datasets. Because vector addition and element-wise multiplication are not sensitive to word order, they perform poorly on the fourteen-choice questions. For both datasets, SuperSim performs significantly

Algorithm	Correct	
	7-choices	14-choices
Vector addition	50.1	22.5
Element-wise multiplication	57.5	27.4
Dual-Space model	58.3	41.5
SuperSim	75.9	68.0
Holistic model	81.6	—

Table 8: Results for the two noun-modifier datasets.

better than all other approaches, except for the holistic approach, according to Fisher's exact test at the 95% confidence level.[8]

The holistic approach is noncompositional. The stem bigram is represented by a single context vector, generated by treating the bigram as if it were a unigram. A noncompositional approach cannot scale up to realistic applications (Turney, 2012). The holistic approach cannot be applied to the fourteen-choice questions, because the bigrams in these questions do not correspond to terms in WordNet, and hence they do not correspond to row vectors in the matrices we use (see Section 3).

Turney (2012) found it necessary to hand-code a soundness check into all of the algorithms (vector addition, element-wise multiplication, dual-space, and holistic). Given a stem ab and a choice c, the hand-coded check assigns a minimal score to the choice if $c = a$ or $c = b$. We do not need to hand-code any checking into SuperSim. It learns automatically from the training data to avoid such choices.

5.2 Ablation Experiments

Table 9 shows the effects of ablating sets of features on the performance of SuperSim with the fourteen-choice questions. PPMI features are the most important; by themselves, they achieve 59.7% correct, although the other features are needed to reach 68.0%. Domain space features reach the second highest performance when used alone (34.6%), but they reduce performance (from 69.3% to 68.0%) when combined with other features; however, the drop is not significant according to Fisher's exact test at the 95% significance level.

Since the PPMI features play an important role in answering the noun-modifier questions, let us take a closer look at them. From Table 2, we see that there are twelve PPMI features for the triple $\langle a, b, c\rangle$, where ab is a noun-modifier bigram and c is a noun unigram. We can split the twelve features into three subsets, one subset for each pair of words, $\langle a, b\rangle$, $\langle a, c\rangle$, and $\langle b, c\rangle$. For example, the subset for $\langle a, b\rangle$ is the four features PPMI$(a, b,$ *left*$)$, PPMI$(b, a,$ *left*$)$, PPMI$(a, b,$ *right*$)$, and PPMI$(b, a,$ *right*$)$. Table 10 shows the effects of ablating these subsets.

Algorithm	Features				Correct
	LF	PPMI	Dom	Fun	
Dual-Space	0	0	1	1	41.5
SuperSim	1	1	1	1	68.0
SuperSim	0	1	1	1	66.6
SuperSim	1	0	1	1	52.3
SuperSim	1	1	0	1	69.3
SuperSim	1	1	1	0	65.9
SuperSim	1	0	0	0	14.1
SuperSim	0	1	0	0	59.7
SuperSim	0	0	1	0	34.6
SuperSim	0	0	0	1	32.9

Table 9: Ablation with fourteen-choice questions.

PPMI feature subsets			
$\langle a, b\rangle$	$\langle a, c\rangle$	$\langle b, c\rangle$	Correct
1	1	1	68.0
0	1	1	59.9
1	0	1	65.4
1	1	0	67.5
1	0	0	62.6
0	1	0	58.1
0	0	1	55.6
0	0	0	52.3

Table 10: PPMI subset ablation with fourteen-choices.

The results in Table 10 indicate that all three PPMI subsets contribute to the performance of SuperSim, but the $\langle a, b\rangle$ subset contributes more than the other two subsets. The $\langle a, b\rangle$ features help to increase the sensitivity of SuperSim to the order of the words in the noun-modifier bigram; for example, they make it easier to distinguish *fantasy world* from *world fantasy*.

5.3 Holistic Training

SuperSim uses 680 training questions to learn to recognize when a bigram is a paraphrase of a unigram; it learns from expert knowledge implicit in WordNet

[8]The results for SuperSim are new but the other results in Table 8 are from Turney (2012).

Stem:		search engine
Choices:	(1)	search_engine
	(2)	search
	(3)	engine
	(4)	search_language
	(5)	search_warrant
	(6)	diesel_engine
	(7)	steam_engine
Solution:	(1)	search_engine

Table 11: A question based on holistic vectors.

synsets. It would be advantageous to be able to train SuperSim with less reliance on expert knowledge.

Past work with adjective-noun bigrams has shown that we can use holistic bigram vectors to train a supervised regression model (Guevara, 2010; Baroni and Zamparelli, 2010). The output of the regression model is a vector representation for a bigram that approximates the holistic vector for the bigram; that is, it approximates the vector we would get by treating the bigram as if it were a unigram.

SuperSim does not generate vectors as output, but we can still use holistic bigram vectors for training. Table 11 shows a seven-choice training question that was generated without using WordNet synsets. The choices of the form a_b are bigrams, but we represent them with holistic bigram vectors; we pretend they are unigrams. We call a_b bigrams *pseudo-unigrams*. As far as SuperSim is concerned, there is no difference between these pseudo-unigrams and true unigrams. The question in Table 11 is treated the same as the question in Table 7.

We generate 680 holistic training questions by randomly selecting 680 noun-modifier bigrams from WordNet as stems for the questions (*search engine*), avoiding any bigrams that appear as stems in the testing questions. The solution (*search_engine*) is the pseudo-unigram that corresponds to the stem bigram. In the matrices in Section 3, each term in WordNet corresponds to a row vector. These corresponding row vectors enable us to treat bigrams from WordNet as if they were unigrams. The distractors are the component unigrams in the stem bigram (*search* and *engine*) and pseudo-unigrams that share a component word with the stem (*search_warrant*, *diesel_engine*). To construct the holistic training questions, we used WordNet as a source of bigrams, but we ignored the rich information that WordNet provides about these bigrams, such as their synonyms, hypernyms, hyponyms, meronyms, and glosses.

	Correct	
Training	7-choices	14-choices
Holistic	61.8	54.4
Standard	75.9	68.0

Table 12: Results for SuperSim with holistic training.

Table 12 compares holistic training to standard training (that is, training with questions like Table 11 versus training with questions like Table 7). The testing set is the standard testing set in both cases. There is a significant drop in performance with holistic training, but the performance still surpasses vector addition, element-wise multiplication, and the hand-coded dual-space model (see Table 8).

Since holistic questions can be generated automatically without human expertise, we experimented with increasing the size of the holistic training dataset, growing it from 1,000 to 10,000 questions in increments of 1,000. The performance on the fourteen-choice questions with holistic training and standard testing varied between 53.3% and 55.1% correct, with no clear trend up or down. This is not significantly different from the performance with 680 holistic training questions (54.4%).

It seems likely that the drop in performance with holistic training instead of standard training is due to a difference in the nature of the standard questions (Table 7) and the holistic questions (Table 11). We are currently investigating this issue. We expect to be able to close the performance gap in future work, by improving the holistic questions. However, it is possible that there are fundamental limits to holistic training.

6 Discussion

SuperSim performs slightly better (not statistically significant) than the hand-coded dual-space model on relational similarity problems (Section 4), but it performs much better on compositional similarity problems (Section 5). The ablation studies suggest this is due to the PPMI features, which have no effect on ten-choice SAT performance (Table 5), but have a

large effect on fourteen-choice noun-modifier paraphrase performance (Table 9).

One advantage of supervised learning over hand-coding is that it facilitates adding new features. It is not clear how to modify the hand-coded equations for the dual-space model of noun-modifier composition (Turney, 2012) to include PPMI information.

SuperSim is one of the few approaches to distributional semantics beyond words that has attempted to address both relational and compositional similarity (see Section 2.3). It is a strength of this approach that it works well with both kinds of similarity.

7 Future Work and Limitations

Given the promising results with holistic training for noun-modifier paraphrases, we plan to experiment with holistic training for analogies. Consider the proportional analogy *hard* is to *hard_time* as *good* is to *good_time*, where *hard_time* and *good_time* are pseudo-unigrams. To a human, this analogy is trivial, but SuperSim has no access to the surface form of a term. As far as SuperSim is concerned, this analogy is much the same as the analogy *hard* is to *difficulty* as *good* is to *fun*. This strategy automatically converts simple, easily generated analogies into more complex, challenging analogies, which may be suited to training SuperSim.

This also suggests that noun-modifier paraphrases may be used to solve analogies. Perhaps we can evaluate the quality of a candidate analogy $\langle a, b, c, d \rangle$ by searching for a term e such that $\langle b, e, a \rangle$ and $\langle d, e, c \rangle$ are good paraphrases. For example, consider the analogy *mason* is to *stone* as *carpenter* is to *wood*. We can paraphrase *mason* as *stone worker* and *carpenter* as *wood worker*. This transforms the analogy to *stone worker* is to *stone* as *wood worker* is to *wood*, which makes it easier to recognize the relational similarity.

Another area for future work is extending SuperSim beyond noun-modifier paraphrases to measuring the similarity of sentence pairs. We plan to adapt ideas from Socher et al. (2011) for this task. They use *dynamic pooling* to represent sentences of varying size with fixed-size feature vectors. Using fixed-size feature vectors avoids the problem of quadratic growth and it enables the supervised learner to generalize over sentences of varying length.

Some of the competing approaches discussed by Erk (2013) incorporate formal logic. The work of Baroni et al. (2012) suggests ways that SuperSim could be developed to deal with logic.

We believe that SuperSim could benefit from more features, with greater diversity. One place to look for these features is higher levels in the hierarchy that we sketch in Section 3.

Our ablation experiments suggest that domain and function spaces provide the most important features for relational similarity, but PPMI values provide the most important features for noun-modifier compositional similarity. Explaining this is another topic for future research.

8 Conclusion

In this paper, we have presented SuperSim, a unified approach to analogy (relational similarity) and paraphrase (compositional similarity). SuperSim treats them both as problems of supervised tuple classification. The supervised learning algorithm is a standard support vector machine. The main contribution of SuperSim is a set of four types of features for representing tuples. The features work well with both analogy and paraphrase, with no task-specific modifications. SuperSim matches the state of the art on SAT analogy questions and substantially advances the state of the art on the SemEval 2012 Task 2 challenge and the noun-modifier paraphrase questions.

SuperSim runs much faster than LRA (Turney, 2006b), answering the SAT questions in minutes instead of days. Unlike the dual-space model (Turney, 2012), SuperSim requires no hand-coded similarity composition functions. Since there is no hand-coding, it is easy to add new features to SuperSim. Much work remains to be done, such as incorporating logic and scaling up to sentence paraphrases, but past work suggests that these problems are tractable.

In the four approaches described by Erk (2013), SuperSim is an instance of the second approach to extending distributional semantics beyond words, comparing word pairs, phrases, or sentences (in general, tuples) by combining multiple pairwise similarity values. Perhaps the main significance of this paper is that it provides some evidence in support of this general approach.

References

Eneko Agirre, Daniel Cer, Mona Diab, 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 (*SEM)*, pages 385–393, Montréal, Canada.

Ion Androutsopoulos and Prodromos Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38:135–187.

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 1183–1193.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, pages 23–32.

Isaac I. Bejar, Roger Chaffin, and Susan E. Embretson. 1991. *Cognitive and Psychometric Analysis of Analogical Problem Solving*. Springer-Verlag.

Ergun Biçici and Deniz Yuret. 2006. Clustering word pairs to answer analogy questions. In *Proceedings of the Fifteenth Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN 2006)*, Akyaka, Mugla, Turkey.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2008. WWW sits the SAT: Measuring relational similarity on the Web. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008)*, pages 333–337, Patras, Greece.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2009. Measuring the similarity between implicit semantic relations from the Web. In *Proceedings of the 18th International Conference on World Wide Web (WWW 2009)*, pages 651–660.

John Bullinaria and Joseph Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526.

John Bullinaria and Joseph Levy. 2012. Extracting semantic representations from word co-occurrence statistics: Stop-lists, stemming, and SVD. *Behavior Research Methods*, 44(3):890–907.

John Caron. 2001. Experiments with LSA scoring: Optimal rank and basis. In *Proceedings of the SIAM Computational Information Retrieval Workshop*, pages 157–169, Raleigh, NC.

Kenneth Church and Patrick Hanks. 1989. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th Annual Conference of the Association of Computational Linguistics*, pages 76–83, Vancouver, British Columbia.

Daoud Clarke. 2012. A context-theoretic framework for compositionality in distributional semantics. *Computational Linguistics*, 38(1):41–71.

Katrin Erk. 2013. Towards a semantics for distributional representations. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)*, Potsdam, Germany.

John Rupert Firth. 1957. A synopsis of linguistic theory 1930–1955. In *Studies in Linguistic Analysis*, pages 1–32. Blackwell, Oxford.

Dan Garrette, Katrin Erk, and Ray Mooney. 2011. Integrating logical representations with probabilistic information using markov logic. In *Proceedings of the 9th International Conference on Computational Semantics (IWCS 2011)*, pages 105–114.

Dedre Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170.

Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. Semeval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007)*, pages 13–18, Prague, Czech Republic.

Emiliano Guevara. 2010. A regression model of adjective-noun compositionality in distributional semantics. In *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics (GEMS 2010)*, pages 33–37.

Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden.

Amaç Herdağdelen and Marco Baroni. 2009. Bagpack: A general framework to represent semantic relations. In *Proceedings of the EACL 2009 Geometrical Models for Natural Language Semantics (GEMS) Workshop*, pages 33–40.

Nathalie Japkowicz and Shaju Stephen. 2002. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449.

David A. Jurgens, Saif M. Mohammad, Peter D. Turney, and Keith J. Holyoak. 2012. SemEval-2012

Task 2: Measuring degrees of relational similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 356–364, Montréal, Canada.

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

Benoît Lemaire and Guy Denhière. 2006. Effects of high-order co-occurrences on word semantic similarity. *Current Psychology Letters: Behaviour, Brain & Cognition*, 18(1).

Yves Lepage and Ando Shin-ichi. 1996. Saussurian analogy: A theoretical account and its application. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING 1996)*, pages 717–722.

Kevin Lund, Curt Burgess, and Ruth Ann Atchley. 1995. Semantic and associative priming in high-dimensional semantic space. In *Proceedings of the 17th Annual Conference of the Cognitive Science Society*, pages 660–665.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2013)*, Atlanta, Georgia.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

Ted Pedersen. 2012. Duluth: Measuring degrees of relational similarity with the gloss vector measure of semantic relatedness. In *First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 497–501, Montreal, Canada.

John C. Platt. 1998. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods: Support Vector Learning*, pages 185–208, Cambridge, MA. MIT Press.

Bryan Rink and Sanda Harabagiu. 2012. UTD: Determining relational similarity using lexical patterns. In *First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 413–418, Montreal, Canada.

Bryan Rink and Sanda Harabagiu. 2013. The impact of selectional preference agreement on semantic relational similarity. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)*, Potsdam, Germany.

Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems (NIPS 2011)*, pages 801–809.

Richard Socher, Brody Huval, Christopher Manning, and Andrew Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012)*, pages 1201–1211.

Mireya Tovar, J. Alejandro Reyes, Azucena Montes, Darnes Vilariño, David Pinto, and Saul León. 2012. BUAP: A first approximation to relational similarity measuring. In *First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 502–505, Montreal, Canada.

Peter D. Turney and Michael L. Littman. 2005. Corpus-based learning of analogies and semantic relations. *Machine Learning*, 60(1–3):251–278.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Peter D. Turney, Michael L. Littman, Jeffrey Bigham, and Victor Shnayder. 2003. Combining independent modules to solve multiple-choice synonym and analogy problems. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-03)*, pages 482–489, Borovets, Bulgaria.

Peter D. Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 680–690.

Peter D. Turney. 2006a. Expressing implicit semantic relations without supervision. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (Coling/ACL-06)*, pages 313–320, Sydney, Australia.

Peter D. Turney. 2006b. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

Peter D. Turney. 2008a. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 33:615–655.

Peter D. Turney. 2008b. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 905–912, Manchester, UK.

Peter D. Turney. 2012. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, 44:533–585.

Tony Veale. 2004. WordNet sits the SAT: A knowledge-based approach to lexical analogy. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, pages 606–612, Valencia, Spain.

Ian H. Witten, Eibe Frank, and Mark A. Hall. 2011. *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition*. Morgan Kaufmann, San Francisco.

Alisa Zhila, Wen-tau Yih, Christopher Meek, Geoffrey Zweig, and Tomas Mikolov. 2013. Combining heterogeneous models for measuring relational similarity. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2013)*, Atlanta, Georgia.