UTD: Determining Relational Similarity Using Lexical Patterns

Bryan Rink and Sanda Harabagiu
University of Texas at Dallas
P.O. Box 830688; MS EC31
Richardson, TX, 75083-0688, USA
{bryan,sanda}@hlt.utdallas.edu

Abstract

In this paper we present our approach for assigning degrees of relational similarity to pairs of words in the SemEval-2012 Task 2. To measure relational similarity we employed lexical patterns that can match against word pairs within a large corpus of 12 million documents. Patterns are weighted by obtaining statistically estimated lower bounds on their precision for extracting word pairs from a given relation. Finally, word pairs are ranked based on a model predicting the probability that they belong to the relation of interest. This approach achieved the best results on the SemEval 2012 Task 2, obtaining a Spearman correlation of 0.229 and an accuracy on reproducing human answers to MaxDiff questions of 39.4%.

1 Introduction

Considerable prior research has examined and elaborated upon a wide variety of semantic relations between concepts along with techniques for automatically discovering pairs of concepts for which a relation holds (Bejar et al., 1991; Stephens and Chen, 1996; Rosario and Hearst, 2004; Khoo and Na, 2006; Girju et al., 2009). However, most previous work has considered membership assignment for a semantic relation as a binary property. In this paper we discuss an approach which assigns a *degree* of membership to a pair of concepts for a given relation. For example, for the semantic relation CLASS-INCLUSION (Taxonomic), the concept pairs *weapon*:*spear* and *bird*:*robin* are stronger members of the relationship than *hair*:*brown*, because *brown* may describe many things other than hair, and brown is also used much less frequently as a noun than the words in the first two word pairs. Task 2 of SemEval 2012 (Jurgens et al., 2012) was designed to evaluate the effectiveness of automatic approaches for determining the similarity of a pair of concepts to a specific semantic relation. The task focused on 79 semantic relations from Bejar et al. (1991) which broadly fall into the ten categories enumerated in Table 1.

Consider the following word pairs: *millionaire:money*, *author:copyright*, *robin:nest*. These X:Y pairs share a relation "X R Y". Now consider the following word pairs:
(1) *teacher:students*
(2) *farmer:crops*
(3) *homeowner:door*
(4) *shrubs:roots*
Which of the numbered word pairs is the MOST illustrative example of the same relation "X R Y"? _____
Which of the above numbered word pairs is the LEAST illustrative example of the same relation "X R Y"? _____

Figure 1: Example Phase 2 MaxDiff question for the relation 2h PART-WHOLE: Creature:Possession.

The data for the task was collected in two phases using Amazon Mechanical Turk [1]. During Phase 1, Turkers were asked to provide pairs of words which fit a relation template, such as "*X* possesses/owns/has *Y*". Turkers provided word pairs such as *expert*:*experience*, *mall*:*shops*, *letters*:*words*, and *doctor*:*degree*. A total of 3,218 word pairs

[1] http://www.mturk.com/mturk/

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 413–418,
Montréal, Canada, June 7-8, 2012.

Category	Example word pairs	Relations
CLASS-INCLUSION	flower:tulip, weapon:knife, clothing:shirt, queen:Elizabeth	5
PART-WHOLE	car:engine, fleet:ship, mile:yard, kickoff:football	10
SIMILAR	car:auto, stream:river, eating:gluttony, colt:horse	8
CONTRAST	alive:dead, old:young, east:west, happy:morbid	8
ATTRIBUTE	beggar:poor, malleable:molded, soldier:fight, exercise:vigorous	8
NON-ATTRIBUTE	sound:inaudible, exemplary:criticized, war:tranquility, dull:cunning	8
CASE RELATIONS	tailor:suit, farmer:tractor, teach:student, king:crown	8
CAUSE-PURPOSE	joke:laughter, fatigue:sleep, gasoline:car, assassin:death	8
SPACE-TIME	bookshelf:books, coast:ocean, infancy:cradle, rivet:girder	9
REFERENCE	smile:friendliness, person:portrait, recipe:cake, astronomy:stars	6

Table 1: The ten categories of semantic relations used in SemEval 2012 Task 2. Each word pair has been taken from a different subcategory of each major category.

across 79 relations were provided by Turkers in Phase 1. Some of these word pairs are naturally more representative of the relationship than others. Therefore, in the second phase, each word pair was presented to a different set of Turkers for ranking in the form of MaxDiff (Louviere and Woodworth, 1991) questions. Figure 1 shows an example MaxDiff question for the relation 2h PART-WHOLE: Creature:Possession ("*X* possesses/owns/has *Y*"). In each MaxDiff question, Turkers were simply asked to select the word pair which was the most illustrative of the relation and the word pair which was the least illustrative of the relation. For the example in Figure 1, most Turkers chose either *shrubs:roots* or *farmer:crops* as the most illustrative of the *Creature:Possession* relation, and *homeowner:door* as the least illustrative. When Turkers select a pair of words they are performing a semantic inference that we wanted to also perform in a computational manner. In this paper we present a method for automatically ranking word pairs according to their relatedness to a given semantic relation.

2 Approach for Determining Relational Similarity

In the vein of previous methods for determining relational similarity (Turney, 2011; Turney, 2008a; Turney, 2008b; Turney, 2005), we propose two approaches using patterns generated from the contexts in which the word pairs occur. Our corpus consists of 8.4 million documents from Gigaword (Parker and Consortium, 2009) and over 4 million articles from Wikipedia. For each word pair, <W1>, <W2> provided by Turkers in Phase 1, as well as the three relation examples, we collected all contexts which matched the schema:

" [0 or more non-content words] <W1> [0 to 7 words] <W2> [0 or more non-content words]"

We also include those contexts where W1 and W2 are swapped. The window size of seven words was determined based on experiments on the training set of ten relations provided by the task organizers. For the non-content words, we considered closed class words such as determiners (*the, who, every*), prepositions (*in, on, instead of*), and conjunctions (*and, but*). Members of these classes were collected from their corresponding Wikipedia pages. Below we provide a sample of the 7,022 contexts found for the word pair *love*:*hate*:

"they <W1> to <W2> it"
"<W1> and <W2> the most . by"
"between <W1> & <W2>"
"<W1> you then i <W2> you and"

We restrict the context before and after the word pair to non-content words in order to match longer contexts without introducing exponential growth in the number of patterns and the consequential sparsity problems. These contexts are directly used as patterns. To generate additional patterns we have one method for shortening contexts and two methods for generating patterns from contexts.

Any contexts which contain words before <W1> or after <W1> are used to create additional shorter contexts by successively removing leading and trailing words. For example, the context "as much <W1> in the <W2> as his" for the word pair *money*:*bank* would generate the following shortened contexts:

"much <W1> in the <W2> as his"
"<W1> in the <W2> as his"

"as much <W1> in the <W2>" as
"as much <W1> in the <W2>"
"much <W1> in the <W2> as"
"<W1> in the <W2> as"
"<W1> in the <W2>"

These shortened contexts are used, along with the original context, to generate patterns.

The first pattern generation method replaces each word between <W1> and <W2> with a wildcard ([^]+ means one or more non-space characters). For example:

"as much <W1> [^]+ the <W2> as"
"as much <W1> in [^]+ <W2> as"

The second pattern generation technique allows for a single word to be matched in the context between the arguments <W1> and <W2>, along with arbitrary matching of other tokens in the context. For example, the context for *red*:*stop* "the <W1> flag is flagged to indicate a <W2>" will generate new patterns such as:

"the <W1>.* flag .*<W2>"
"the <W1>.* is .*<W2>"
"the <W1>.* flagged .*<W2>"
"the <W1>.* indicate .*<W2>"

After all patterns have been generated, they are used by our two approaches to assign relational similarity scores to word pairs.

2.1 UTD-NB Approach

The first of our two approaches, UTD-NB, assigns weights to patterns which are then used to assign similarity scores to word pairs. The approach begins by obtaining all word pairs associated with a relation. Each relation is associated with a target set (T) of word pairs from two sources: (i) the three or four example word pairs provided for each relation, and (ii) the word pairs provided by Turkers in Phase 1. We collect all of the contexts for those word pairs to generate patterns. The UTD-NB approach assumes that the word pairs provided by Turkers, while noisy, can be used to characterize the relation. As an example, consider these word pairs provided by Turkers for the relation 8a (Cause:Effect) *illness:discomfort*, *fire:burns*, *accident:damage*. A pattern which extracts these word pairs is: "<W1> that caused [^]+ <W2>". This pattern is unlikely to match the contexts of word pairs from other relations. Therefore, we use the statistics about how many target word pairs a pattern extracts versus how many non-target pairs a pattern extracts to assign a weight to the pattern. A pattern which matches many of the word pairs from the target relation and few (or none) of the word pairs from other relations is likely to be a good indicator of that relation. For example, the pattern **P1** for the relation 8a (Cause:Effect): "the <W1>.* caused .*<W2> to his" matches only three word pairs: *explosion*:*damage*, *accident*:*damage*, and *injury*:*pain*, all of them belonging to the target relation. Conversely, the pattern **P2**: "<W1>.* causing .*<W2> but" matches five words pairs. However, only three of them belong to the target relation: *hit*:*injury*, *explosion*:*damage*, *germs*:*sickness*. The remaining two: *city*:*people*, *action*:*alarm* belong to other relations: .

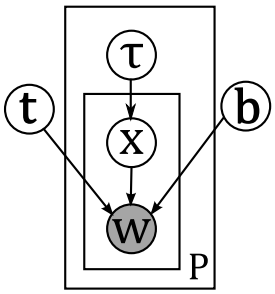

Figure 2: Probabilistic model for the word pairs extracted by patterns, for a single relation.

We use the number of target word pairs extracted, x, and the total number of word pairs extracted, n, to calculate τ: the probability that a word pair extracted by the pattern will belong to the target relation. The maximum likelihood estimate for τ is $\frac{x}{n}$, however for small values of x this estimate has a high variance and can significantly overestimate the true value. Therefore, we used the Wilson interval score for determining a lower bound on τ at a 99.9% confidence level. This gives the pattern **P1** above with $x = 3$ and $n = 3$ a lower bound on τ of 21.7% and **P2** with $x = 3$ and $n = 5$ a lower bound on τ of 16.6%. We use this lower bound as the pattern's weight. These pattern weights are then combined to score each word pair for the target relation.

We model the word pairs extracted by the patterns as a generative process shown in Figure 2. Each pattern, p, is associated with with a precision, τ, which is the probability that a word pair extracted by that pattern is a member of the target relation. The ob-

served word pairs extracted by a pattern are denoted by w. Our model assumes that a word pair extracted by a pattern may be drawn from one of two distinct distributions over word pairs: a distribution for the target relation $\vec{t}$, and a background distribution over word pairs $\vec{b}$. The generation of a word pair begins with a binary variable x drawn from a Bernoulli distribution parametrized by τ (the pattern's precision), which represents whether a word pair is generated according to a relation specific distribution, or a background distribution. More explicitly, if $x = 1$, then a word pair w is generated by the target relation distribution $\vec{t}$, and if $x = 0$, a word pair is generated by the background distribution $\vec{b}$.

We may not yet perform any meaningful inference because no evidence has been observed to correctly infer whether the target distribution or the background distribution generated w. Therefore we use the pattern weights derived above (based on the lower bounds on the pattern precisions) as that pattern's value of τ. For estimating the distributions $\vec{t}$ and $\vec{b}$, we assume that x is 1 (w is generated by $\vec{t}$) if and only if $\tau \geq 0.1$ and the word pair w belongs to the target set of word pairs T. This threshold on τ has a filtering effect on the patterns, and those patterns below the threshold are treated as non-indicative of the relation. These assumptions allow us to estimate the parameters for $\vec{t}$ and $\vec{b}$:

$$P(w|\vec{t}) = \begin{cases} \frac{\#(w,h)}{\#(h)} & \text{if } w \in T \\ 0 & \text{if } w \notin T \end{cases} \quad (1)$$

$$P(w|\vec{b}) = \frac{\#(w, \neg h) + \#(w,h)\mathbf{1}_{w \notin T}}{\sum_u \#(u, \neg h) + \#(u,h)\mathbf{1}_{u \notin T}} \quad (2)$$

where $\#(w,h)$ is the number of times w was extracted by a high precision pattern ($\tau \geq 10\%$), and $\#(h)$ is the number of word pairs extracted by a high precision pattern.

The only remaining hidden variable in the model is x which we can now estimate using the inferred distributions for the other variables. We chose to use the probability of x for a word pair w as the score by which we rank the word pairs. Furthermore, we use only the probability of x for the highest ranking pattern p which extracted w:

$$P(x = 1|p, w) = \frac{P(x = 1, w|p)}{P(w|p)} \quad (3)$$

where $P(x = 1, w|p) = \tau_p \times \vec{t}(w)$ and $P(w|p) = P(x = 1, w|p) + P(x = 0, w|p)$

This method of scoring word pairs accounts for how common a word pair is overall. For example for the relation 4c (CONTRAST: Reverse), the word pair *white*:*black* occurs very commonly in both high precision patterns and low precision patterns (those more likely associated with other relations). Therefore even though the word pair shares its highest ranking pattern with the pair *eat*:*fast*, *white*:*black* receives a score of 0.019 while *eat*:*fast* receives a score of 0.216 because $\vec{t}(white : black) = 0.006$ and $\vec{b}(white : black) = 0.104$, while $\vec{t}(eat : fast) = 0.0016$ and $\vec{b}(eat : fast) = 0.0018$. However, if a pattern with 100% precision were to extract *white*:*black*, the pair would appropriately receive a score of 1.0 despite being much more common in the background distribution. This is motivated by our assumption that such a pattern can only extract word pairs which truly belong to the relation. Another motivation for scoring word pairs by their highest ranking pattern is that it does not depend on any assumption of independence between the patterns which extract the pairs. For example, the pattern "<W1> , not <W2> . " extracts largely the same word pairs as "<W1> [^]+ not <W2> ." and thus its matches should not be taken as additional evidence about the word pairs.

2.2 UTD-SVM Approach

Our second approach uses an SVM-rank (Joachims, 2006) model to rank the word pairs. Each word pair from a target relation is represented as a binary feature vector indicating which patterns extracted the word pair. We train the SVM-rank classifier by assigning all word pairs from the target relation rank 2, and all word pairs from other relations with rank 1. The SVM model is then trained and used to classify the word pairs from the target relation. Even though the model is used to classify the same word pairs it was trained on, it still provides higher scores to word pairs more likely to belong to the target relation. We directly rank the word pairs using these scores.

3 Discussion

The organizers of SemEval 2012 Task 2 viewed relational similarity in two different ways. The first

Word pair	% Most illustrative - % Least illustrative
"freezing:warm"	56.0
"earsplitting:quiet"	36.0
"evil:angelic"	18.0
"ancient:modern"	12.0
"disastrous:peaceful"	6.0
"ecstatic:disgruntled"	2.0
"disgusting:tasty"	0.0
"beautiful:plain"	-2.0
"dirty:sterile"	-4.0
"wrinkled:smooth"	-6.0
"sweet:sour"	-20.0
"disgruntled:ecstatic"	-32.0
"white:gray"	-54.0

Table 2: A sample of the 41 word pairs provided by Amazon Mechanical Turk participants for the relation 4f (CONTRAST: Asymmetric Contrary - X and Y are at opposite ends of the same scale). The word pairs are ranked by how illustrative of the relation participants found each pair to be.

view was that of solving a MaxDiff problem, question in which participants are shown a list of four word pairs and asked to select the most and least illustrative pairs. The second view of relation similarity considers the task of assigning scores to a according to their similarity to the relation of interest. The first column of Table 2 provides an example of word pairs that Amazon Turkers said belonged to the 4f: CONTRAST: *Asymmetric Contrary* relation in Phase 1, ranked according to how well other Turkers felt they represented the relation. The score in the second column is calculated as the percentage of how often Turkers rated a word pair as the most illustrative and how often Turkers rated the word pair as the least illustrative.

Both of our approaches for determining relation similarity assign scores directly to the word pairs collected in Phase 1, with the goal of ranking the words in the same order that was induced from the responses by Amazon Mechanical Turkers.

3.1 Evaluation Measures

SemEval-2012 Task 2 had two official evaluation metrics. The first directly measured the accuracy of automatically choosing the most and least illustrative word pairs among a set of four word pairs taken from responses during Phase 1. The accuracy of choosing the most illustrative word pair and the accuracy of choosing the least illustrative word pair were calculated separately and averaged to produce the MaxDiff accuracy.

Team-Algorithm	Spearman	MaxDiff
UTD-NB	0.229	39.4
UTD-SVM	0.116	34.7
Duluth-V0	0.050	32.4
Duluth-V1	0.039	31.5
Duluth-V2	0.038	31.1
BUAP	0.014	31.7
Random	0.018	31.2

Table 3: Results for all systems participating in SemEval 2012 Task 2 on relational similarity, including a random baseline.

The second evaluation metric measured the correlation between an automatic ranking of word pairs for a relation and a ranking induced by the Turkers' responses to the MaxDiff questions. The word pairs were given scores equal to the percentage of times they were chosen by Turkers as the most illustrative example for a relation minus the percentage of times they were chosen as the least illustrative. Systems were then evaluated according to their Spearman rank correlation with the ranking of word pairs induced by that score. Spearman correlations range from -1 for a negative correlation to 1.0 for a perfect correlation.

3.2 Results

Table 3 shows the results for the six systems which participated in SemEval-2012 Task 2, along with the results for a baseline which ranks each word pair randomly. Our two approaches achieved the best results on both evaluation metrics. Our UTD-NB approach achieves much better performance than our UTD-SVM approach, likely due to the unconventional use of the SVM to classify its own training data. That said, the results are still significantly higher than those of other participants. This may be attributed to our incorporation of better patterns or our use of a large corpus. It might also be a consequence of our approaches considering all of the testing word pairs simultaneously.

Table 4 shows the results for each of the ten categories of relations. The best results are achieved on SPACE-TIME relations, while the lowest performance is on the NON-ATTRIBUTE relations. NON-

Category	Rndm	BUAP	UTD NB	UMD V0
1 CLASS-INCLUSION	0.057	0.064	0.233	0.045
2 PART-WHOLE	0.012	0.066	0.252	-0.061
3 SIMILAR	0.026	-0.036	0.214	0.183
4 CONTRAST	-0.049	0.000	0.206	0.142
5 ATTRIBUTE	0.037	-0.095	0.158	0.044
6 NON-ATTRIBUTE	-0.070	0.009	0.098	0.079
7 CASE RELATIONS	0.090	-0.037	0.241	-0.011
8 CAUSE-PURPOSE	-0.011	0.114	0.183	0.021
9 SPACE-TIME	0.013	0.035	0.375	0.055
10 REFERENCE	0.142	-0.001	0.346	0.028

Table 4: Spearman correlation results for the best system from each team, across all ten categories of relations.

ATTRIBUTE relations associate objects and actions with an atypical attribute (*harmony*:*discordant*, *immortal*:*death*, *recluse*:*socialize*). Because the pairs of words associated with these relation are not typically associated together, our approach likely performs poorly on these relations because our approach is based on finding the pairs of words together in a large corpus.

An interesting consequence of the 10% precision threshold used in the UTD-NB approach is that 24 relations had no patterns exceeding the threshold and therefore produced zeroes as scores for all word pairs. However, word pairs which never occurred within seven tokens of each other in our corpus received a negative score and were ranked lower. Such rankings tend to produce Spearman scores around 0.0. Our lowest Spearman score was -0.068, while other teams had low scores of -0.344 and -0.266, both occurring on relations for which UTD-NB produced no positive word pair scores. There are two lessons to be learned from this result: (i) the UTD-NB approach does a good job of recognizing when it cannot rank word pairs, and (ii) such relations are likely difficult and worth further investigation.

4 Conclusion

We described the UTD approaches to determining relation similarity using lexical patterns from a large corpus. Combined with a probabilistic model for word pair extraction by those patterns, we were able to achieve the highest performance at the SemEval 2012 Task 2. Our results showed the approach significantly outperformed a model which used an SVM-rank model used to classify its own training set. The approach also performed well across a wide range of relation types and argument classes which included nouns, adjectives, verbs, and adverbs. This implies that the approaches presented in this paper could be successfully applied to other domains which involve semantic relations.

References

Isaac I. Bejar, Roger Chaffin, and Susan E. Embretson. 1991. *Cognitive and psychometric analysis of analogical problem solving. Recent research in psychology.* Springer-Verlag Publishing.

Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2009. Classification of semantic relations between nominals. *Language Resources and Evaluation*, 43(2):105–121.

Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference KDD '06*, page 217, New York, New York, USA, August. ACM Press.

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 6th International Workshop on Semantic Evaluation (SemEval 2012)*.

Christopher S G Khoo and Jin-cheon Na. 2006. Semantic relations in information science. *Annual Review of Information Science and Technology*, 40(1):157–228.

Jordan J Louviere and G G Woodworth. 1991. Best-worst scaling: A model for the largest difference judgments. Technical report, University of Alberta.

Robert Parker and Linguistic Data Consortium. 2009. *English gigaword fourth edition*. Linguistic Data Consortium.

Barbara Rosario and Marti A. Hearst. 2004. Classifying semantic relations in bioscience texts. In *Proceedings of the AACL '04*, pages 430–es, July.

Larry M. Stephens and Yufeng F. Chen. 1996. Principles for organizing semantic relations in large knowledge bases. *IEEE Transactions on Knowledge and Data Engineering*, 8(3):492–496, June.

Peter D. Turney. 2005. Measuring Semantic Similarity by Latent Relational Analysis. In *International Joint Conference On Artificial Intelligence*, volume 19.

Peter D. Turney. 2008a. A Uniform Approach to Analogies, Synonyms, Antonyms, and Associations. In *Proceedings of COLING '08*, August.

Peter D. Turney. 2008b. The Latent Relation Mapping Engine: Algorithm and Experiments. *Journal of Artificial Intelligence Research*, 33:615–655.

Peter D. Turney. 2011. Analogy perception applied to seven tests of word comprehension. *Journal of Experimental & Theoretical Artificial Intelligence*, 23(3):343–362, July.