Abstract

This paper describes and analyzes our SemEval 2014 Task 1 system. Its features are based on distributional and denotational similarities; word alignment; negation; and hypernym/hyponym, synonym, and antonym relations.

1 Task Description

SemEval 2014 Task 1 (Marelli et al., 2014a) evaluates system predictions of semantic relatedness (SR) and textual entailment (TE) relations on sentence pairs from the SICK dataset (Marelli et al., 2014b). The dataset is intended to test compositional knowledge without requiring the world knowledge that is often required for paraphrase classification or Recognizing Textual Entailment tasks. SR scores range from 1 to 5. TE relations are ‘entailment,’ ‘contradiction,’ and ‘neutral.’

Our system uses features that depend on the amount of word overlap and alignment between the two sentences, the presence of negation, and the semantic similarities of the words and substrings that are not shared across the two sentences. We use simple distributional similarities as well as the recently proposed denotational similarities of Young et al. (2014), which are intended as more precise metrics for tasks that require entailment. Both similarity types are estimated on Young et al.’s corpus, which contains 31,783 images of everyday scenes, each paired with five descriptive captions.

2 Our system

Our system combines different sources of semantic similarity to predict semantic relatedness and textual entailment. We use distributional similarity features, denotational similarity features, and alignment features based on shallow syntactic structure.

2.1 Preprocessing

We lemmatize all sentences with the Stanford CoreNLP system1 and extract syntactic chunks with the Illinois Chunker (Punyakanok and Roth, 2001). Like Young et al. (2014), we use the Malt parser (Nivre et al., 2006) to identify 5 sets of constituents for each sentence: subject NPs, verbs, VPs, direct object NPs, and other NPs.

For stopwords, we use the NLTK English stopword list of 127 high-frequency words. We remove negation words (no, not, and nor) from the stopword list since their presence is informative for this dataset and task.

2.2 Distributional similarities

After stopword removal and lemmatization, we compute vectors for tokens that appear at least 10 times in Young et al. (2014)’s image description corpus. In the vector space, each dimension corresponds to one of the 1000 most frequent lemmas (contexts). The jth entry of the vector of wi is the positive normalized pointwise mutual information (pnPMI) between target wi and context wj:

\[
pnPMI(w_i, w_j) = \max \left( 0, \frac{\log \left( \frac{P(w_i, w_j)}{P(w_i)P(w_j)} \right)}{-\log(P(w_i, w_j))} \right)
\]

We define \( P(w_i) \) as the fraction of images with at least one caption containing \( w_i \), and \( P(w_i, w_j) \) as the fraction of images whose captions contain both \( w_i \) and \( w_j \). Following recent work that extends distributional similarities to phrases and sentences (Mitchell and Lapata, 2010; Baroni and Zamparelli, 2010; Grefenstette and Sadrzadeh, 2011; Socher et al., 2012), we define a phrase vector \( p \) to be the pointwise multiplication product of the vectors of the words in the phrase:

\[
p = w_1 \odot \ldots \odot w_n
\]

1http://nlp.stanford.edu/software/corenlp.shtml
Negation In this dataset, contradictory sentence
pairs are often marked by explicit negation, e.g. $s_1 = \text{"The man is stirring the sauce for the chicken"}$ and $s_2 = \text{"The man is not stirring the sauce for the chicken."}$ A binary feature is set to 1 if either sentence contains not, no, or nobody, and set to 0 otherwise.

Word overlap We compute $|W_1 \setminus W_2|$ on lemmatized sentences without stopwords where $W_i$ is the set of word types that appear in $s_i$. Training a MaxEnt or log-linear model using this feature achieves better performance than the word overlap baseline provided by the task organizers.

Denotational constituent similarity Denotational similarity captures entailment-like relations between events. For example, sit and eat lunch have a high pnPMI, which follows our intuition that a person who is eating lunch is likely to be sitting. We use the same denotational constituent features that Young et al. (2014) use for a textual similarity task. $C$ are original nodes, $C^{\text{anc}}$ are parent and grandparent nodes, and $\text{sim}(C_a, C_b)$ is the maximum pnPMI of any pair of nodes $a \in C_a$, $b \in C_b$.

C-C features compare constituents of the same type. These features express how often we expect corresponding constituents to describe the same situation. For example, $s_1 = \text{"Girls are doing backbends and playing outdoors"}$ and $s_2 = \text{"Children are doing backbends"}$ have subject nodes $\{\text{girl}\}$ and $\{\text{child}\}$. Girls are sometimes described as children, so $\text{sim(\text{girl}, \text{child})} = 0.498$. In addition, child is a parent node of girl, so $\text{max(\text{sim(anc(girl), child)}) = 1}$. There are 15

| Features                      | Description                                                                 | # of features |
|-------------------------------|-----------------------------------------------------------------------------|---------------|
| Negation                      | True if either sentence contains explicit negation; False otherwise         | 1             |
| Word overlap                  | Ratio of overlapping word types to total word types in $s_1$ and $s_2$      | 1             |
| Denotational constituent sim. | Positive normalized PMI of constituent nodes in the denotation graph        | 30            |
| Distributional constituent sim.| Cosine similarity of vector representations of constituent phrases          | 30            |
| Alignment                     | Ratio of number of aligned words to length of $s_1$ and $s_2$; max, min,   | 23            |
|                               | average unaligned chunk length; number of unaligned chunks                 |               |
| Unaligned matching            | Ratio of number of matched chunks to unaligned chunks; max, min,           | 31            |
|                               | average matched constituent similarity; number of crossings in matching      |               |
| Chunk alignment               | Number of chunks; number of unaligned chunk labels; ratio of unaligned     | 17            |
|                               | chunk labels to number of chunks; number of matched labels; ratio of       |               |
|                               | matched to unmatched chunk labels                                         |               |
| Synonym                       | Number of matched synonym pairs $(w_1, w_2)$                               | 1             |
| Hypernym                      | Number of matched hypernym pairs $(w_1, w_2)$, number of matched           | 2             |
|                               | hypernym pairs $(w_2, w_1)$                                                |               |
| Antonym                       | Number of matched antonym pairs $(w_1, w_2)$                               | 1             |

Table 1: Summary of features
C-C features: $\text{sim}(C_1, C_2)$, $\max(\text{sim}(C_1, C_2^{\text{unc}}),$ $\text{sim}(C_1^{\text{unc}}, C_2))$, $\min(\text{sim}(C_1^{\text{unc}}, C_2^{\text{unc}}))$ for each constituent type.

C-all features compare different constituent types. These features express how often we expect any pair of constituents to describe the same scene. For example, $s_1 = \text{"Two teams are competing in a football match"}$ and $s_2 = \text{"A player is throwing a football"}$ are topically related sentences. Comparing constituents of different types like player and compete or player and football match gives us more information about the similarity of the sentences. There are 15 C-all features: the maximum, minimum, and sum of $\text{sim}(C_1, C_2)$ and $\text{sim}(C_1, C_2^2)$ for each constituent type.

**Distributional constituent similarity** Distributional vector-based similarity may alleviate the sparsity of the denotation graph. For example, for subject NP C-C features, we have non-zero distributional similarity for 87% of instances in the trial data, but non-zero denotational similarity for only 56% of the same instances. The football and team nodes may have no common images in the denotation graph, but we still have distributional vectors for football and for team. The 30 distributional similarity features are the same as the denotational similarity features except $\text{sim}(a, b)$ is the cosine similarity between constituent phrase vectors.

**Alignment** Since contradictory and entailling sentences have limited syntactic variation in this dataset, aligning sentences can help to predict semantic relatedness and textual entailment. We use the Needleman-Wunsch algorithm (1970) to compute an alignment based on exact word matches between two lemmatized sentences. The similarity between two lemmas is 1.0 if the words are identical and 0.0 otherwise, and we do not penalize gaps. This gives us the longest subsequence of matching lemmas.

The alignment algorithm results in a sentence pair alignment and 2 unaligned chunk sets defined by syntactic chunks. For example, $s_1 = \text{"A brown and white dog is running through the tall grass"}$ and $s_2 = \text{"A brown and white dog is moving through the wild grass"}$ are mostly aligned, with the remaining chunks $u_1 = \{\text{[VP run]}, \text{[NP tall]}\}$ and $u_2 = \{\text{[VP move]}, \text{[NP wild]}\}$.

There are 23 alignment features. Directional features per sentence are the number of words (2 features), the number of aligned words (2 features), and the ratio between those counts (2 features). These features are expressed twice, once according to the sentence order in the dataset and once ordered by longer sentence before shorter sentence, for a total of 12 directional features. Non-directional features are the maximum, minimum, and average unaligned chunk length for each sentence and for both sentences combined (9 features), and the number of unaligned chunks in each sentence (2 features).

**Unaligned chunk matching** We want to know the similarity of the remaining unaligned chunks because when two sentences have a high overlap, their differences are very informative. For example, in the case that two sentences are identical except for a single word in each sentence, if we know that the two words are synonymous, then we should predict that the two sentences are highly similar. However, if the two words are antonyms, the sentences are likely to be contradictory.

We use phrase vector similarity to compute the most likely matches between unaligned chunks. We repeat the matching process twice: for simple matching, any 2 chunks with non-zero phrase similarity can be matched across sentences, while for strict matching, chunks can match only if they have the same type, e.g. NP or VP. This gives us two sets of features.

For $s_1 = \text{"A brown and white dog is running through the tall grass"}$ and $s_2 = \text{"A brown and white dog is moving through the wild grass"}$, the unaligned chunks are $u_1 = \{\text{[VP run]}, \text{[NP tall]}\}$ and $u_2 = \{\text{[VP move]}, \text{[NP wild]}\}$. For strict matching, the only valid matches are $\text{[VP run]} - \text{[VP move]}$ and $\text{[NP tall]} - \text{[NP wild]}$. For simple matching, $\text{[NP tall]}$ could also match $\text{[VP move]}$ instead and $\text{[VP run]}$ could match $\text{[NP wild]}$.

There are a total of 31 unaligned chunk matching features. Directional features per sentence include the number of unaligned chunks (2 features) and the ratio of the number of matched chunks to the total number of chunks (2 features). These features are expressed twice, once according to the sentence order in the dataset and once ordered by longer sentence before shorter sentence, for a total of 8 directional features. Non-directional features per sentence pair include
the maximum, minimum, and average similarity of the matched chunks (3 features); the maximum, minimum, and average length of the matched chunks (3 features); and the number of matched chunks (1 feature). We extract these 15 features for both simple matching and strict matching. In addition, we also count the number of crossings that result from matching the unaligned chunks in place (1 feature). This penalizes matched sets that contain many crossings or long-distance matches.

**Chunk label alignment and matching** Since similar sentences in this dataset often have similar syntax, we compare their chunk label sequences, e.g. [NP A brown and white dog] [VP is running] [PP through] [NP the tall grass] becomes NP VP PP NP. We compute 17 features based on aligning and matching these chunk label sequences. Directional features are the total number of labels in the sequence (2 features), the number of unaligned labels (2 features), the ratio of the number of unaligned labels to the total number of labels (2 features), and the ratio of the number of matched labels to the number of unaligned labels (2 features). These features are expressed twice, once according to the sentence order in the dataset and once ordered by longer sentence before shorter sentence, for a total of 16 directional features. We also count the number of matched labels for the sentence pair (1 feature).

**Synonyms and Hypernyms** We count the number of synonyms and hypernyms in the matched chunks for each sentence pair. Synonyms are words that share a WordNet synset, and hypernyms are words that have a hypernym relation in WordNet. There are two hypernym features because hypernymy is directional: \( \text{num}_{\text{hyp}}_1 \) is the number of words in \( s_1 \) that have a hypernym in \( s_2 \), while \( \text{num}_{\text{hyp}}_2 \) is the number of words in \( s_2 \) that have a hypernym in \( s_1 \). For example, \( s_1 = \text{"A woman is cutting a lemon"} \) and \( s_2 = \text{"A woman is cutting a fruit"} \) have \( \text{num}_{\text{hyp}}_1 = 1 \).

For synonyms, \( \text{num}_{\text{syn}} \) is the number of word pairs in \( s_1 \) and \( s_2 \) that are synonyms. For example, \( s_1 = \text{"A brown and white dog is running across the lawn"} \) and \( s_2 = \text{"A big white dog is running across a lawn"} \). For example, \( \text{num}_{\text{syn}} = 1 \).

**Antonyms** When we match unaligned chunks, the highest similarity pair are sometimes antonyms, e.g. \( s_1 = \text{"Some people are on a crowded street"} \) and \( s_2 = \text{"Some people are on an empty street."} \). In other cases, they are terms that we think of as mutually exclusive, e.g. man and woman. In both cases, the sentences are unlikely to be in an entailing relationship. Since resources like WordNet will fail to identify the mutually exclusive pairs that are common in this dataset, e.g. bike and car or piano and guitar, we use the training data to build a list of these pairs. We identify the matched chunks that occur in contradictory or neutral sentences but not entailed sentences. We exclude synonyms and hypernyms and apply a frequency filter of \( n = 2 \). Commonly matched chunks in neutral or contradictory sentences include sit–stand, boy–girl, and cat–dog. These are terms with different and often mutually exclusive meanings. Commonly matched chunks in entailed sentences include man–person, and lady–woman. These are terms that could easily be used to describe the same situation. However, cut–slice is a common pair in both neutral and entailed sentences and we do not want to count it as an antonym pair. Therefore, we consider frequent pairs that occur in contradictory or neutral but not entailed sentences to be antonyms.

The feature \( \text{num}_{\text{ant}} \) is the number of matched antonyms in a sentence pair. We identify an antonym if \( c_a \) and \( c_b \) are on the antonym list or occur in one of these patterns: \( X\text{–}not \ X, \ X\text{–}no \ X, \ X\text{–}no \ head\text{-noun}(X) \) (e.g. blue hat–no hat), \( X\text{–}no \ hypernym(X) \) (e.g. poodle–no dog), \( X\text{–}no \ synonym(X) \) (e.g. kid–no child). For each antonym pair, we set the similarity score of that match to 0.0.

For example, \( \text{num}_{\text{ant}} = 1 \) for \( s_1 = \text{"A small white dog is running across a lawn"} \) and \( s_2 = \text{"A big white dog is running across a lawn."} \). In addition, \( \text{num}_{\text{ant}} = 1 \) for \( s_1 = \text{"A woman is leaning on the ledge of a balcony"} \) and \( s_2 = \text{"A man is leaning on the ledge of a balcony."} \).

2.5 Models

For the SR task, we implement a log-linear regression model using Weka (Hall et al., 2009). Specifically, under Weka’s default settings, we train a ridge regression model with regularization parameter \( \alpha = 1 \times 10^{-8} \). For the TE task, we use a MaxEnt model implemented with MALLET (McCallum, 2002). The MaxEnt model is optimized with
Table 2: TE and SR results on test data

| Model                  | Accuracy | Pearson ρ |
|------------------------|----------|-----------|
| Overlap baseline       | 56.8     | 0.646     |
| Negation               | 61.0     | 0.093     |
| Word overlap (+Vector composition) | 66.4 | 0.697 |
| +Denotational similarity | 74.4      | 0.751     |
| +Distributional similarity | 71.8      | 0.756     |
| +Den +Dist              | 77.0     | 0.782     |
| +Alignment              | 70.4     | 0.697     |
| +Unaligned chunk matching | 75.8      | 0.719     |
| +Align +Match           | 75.2     | 0.728     |
| +Synonyms               | 65.2     | 0.696     |
| +Hypernyms              | 66.8     | 0.716     |
| +Antonyms               | 71.0     | 0.704     |
| All features            | 84.2     | 0.802     |

Table 3: TE and SR results on trial data

| Model                  | Accuracy | Pearson ρ |
|------------------------|----------|-----------|
| Overlap baseline       | 77.3     | 44.8      |
| Negation               | 85.4     | 0.0       | **86.4** |
| Word overlap (+Vector composition) | 84.7 | 64.5 |
| +Denotational similarity | 83.6     | 67.3      |
| +Distributional similarity | 86.5     | 60.4      |
| +Den +Dist              | 85.4     | 68.7      |
| +Alignment              | 87.9     | 50.6      |
| +Unaligned chunk matching | **90.4**  | 66.6      |
| +Align +Match           | 88.6     | 61.8      |
| +Synonyms               | 82.2     | 65.2      |
| +Hypernyms              | 84.0     | 68.0      |
| +Antonyms               | 83.6     | 82.6      |
| All features            | 86.5     | **83.3**  |

Table 4: TE accuracy on trial data by entailment type (Neutral, Entailment, Contradiction)

| Model                          | N  | E  | C  |
|--------------------------------|----|----|----|
| Overlap baseline               | 77.3| 44.8| 0.0 |
| Negation                       | 85.4| 0.0 | **86.4** |
| Word overlap (+Vector composition) | 84.7 | 64.5 |
| +Denotational similarity       | 83.6| 67.3| 52.7|
| +Distributional similarity     | 86.5| 60.4| 37.8|
| +Den +Dist                     | 85.4| 68.7| 60.8|
| +Alignment                     | 87.9| 50.6| 41.8|
| +Unaligned chunk matching     | **90.4**| 66.6| 37.8|
| +Align +Match                  | 88.6| 61.8| 50.0|
| +Synonyms                      | 82.2| 65.2| 0.0 |
| +Hypernyms                     | 84.0| 68.0| 0.0 |
| +Antonyms                      | 83.6| 82.6| 0.0 |
| All features                   | 86.5| **83.3**| 77.0|

3 Results

Our submitted system was trained on the full training and trial data (5000 sentences). Table 2 shows our results on the test data. We substantially outperform all baselines.

3.1 Feature Ablation

We analyze the performance of our features on the trial data using models trained only on the training data. Models marked with + include our word overlap feature. We also examine a single compositional feature (vector composition), which is the cosine similarity of the two sentence vectors. A sentence vector is the pointwise multiplication product of its component word vectors.

Table 3 compares these models on both tasks. For TE, unaligned chunk matching outperforms the other features. Denotational constituent similarity does almost as well. For SR, distributional and denotational features have the highest correlation with the gold scores, and combining them improves performance even more.

Table 4 shows the TE accuracy of each model by entailment label. The negation model correctly classifies 86.0% of contradictions while our final system has only 77.0% accuracy on contradic-

4 Conclusion

Our system combines multiple similarity metrics to predict semantic relatedness and textual entailment. Features that identify negation and make similarity comparisons based on chunking do very well. Denotational constituent similarity features also show strong performance on both tasks. In the future, we would like to focus on multiword paraphrases and prepositional phrases, which our current system has trouble analyzing.

Acknowledgements

We gratefully acknowledge support of the National Science Foundation under grants 1053856 and 1205627, as well as an NSF Graduate Research Fellowship to Alice Lai.
References

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*, pages 1183–1193, Cambridge, MA, October. Association for Computational Linguistics.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1):10–18.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaela Bernardi, Stefano Menini, and Roberto Zamparelli. 2014a. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and text entailment. In *Proceedings of SemEval 2014: International Workshop on Semantic Evaluation*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaela Bernardi, and Roberto Zamparelli. 2014b. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of LREC*.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

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

Saul B. Needleman and Christian D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443 – 453.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC*, volume 6, pages 2216–2219.

Vasin Punyakanok and Dan Roth. 2001. The use of classifiers in sequential inference. In *NIPS*, pages 995–1001. MIT Press.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. 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*, pages 1201–1211, Jeju Island, Korea, July. Association for Computational Linguistics.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.