A Logic Prover Approach to Predicting Textual Similarity

Eduardo Blanco and **Dan Moldovan**
Lymba Corporation
Richardson, TX 75080
{*eduardo,moldovan*}*@lymba.com*

Abstract

This paper presents a logic prover approach to predicting textual similarity. Sentences are represented using three logic forms capturing different levels of knowledge, from only content words to semantic representations extracted with an existing semantic parser. A logic prover is used to find proofs and derive semantic features that are combined in a machine learning framework. Experimental results show that incorporating the semantic structure of sentences yields better results than simpler pairwise word similarity measures.

1 Introduction

The task of Semantic Textual Similarity (Agirre et al. 2012) measures the degree of semantic equivalence between two sentences. Unlike textual entailment (Giampiccolo et al. 2007), textual similarity is symmetric, and unlike both textual entailment and paraphrasing (Dolan and Brockett 2005), textual similarity is modeled using a graded score rather than a binary decision. For example, sentence pair (1) is very similar, while (2) is somewhat similar and (3) is not similar.

1. A man is riding a bicycle. A man is riding a bike.
2. A man is cutting a paper. A person is tearing paper.
3. A car is parking. A cat is playing.

State-of-the-art systems that predict textual similarity (Bär et al. 2012; Šarić et al. 2012; Banea et al. 2012) mostly rely on word pairings and disregard the semantic structure of sentences. Consider sentences 1(a) *A man is holding a leaf* and 1(b) *A monkey is fighting with a man*. These two sentences are very dissimilar; the only commonality is the concept '*man*'. Any approach that blindly searches for the word in 1(b) that is the most similar to the word '*man*' in 1(a) will find '*man*' from 1(b) to be a match. One of three content words is a perfect match and thus the predicted similarity will be much higher than it actually is.

Consider now the semantic representations for sentences 1(a) and 1(b) in Figure 1. '*man*' is an AGENT in 1(a), and a THEME in 1(b). While both words encode the same concept, their semantic functions with respect to other concepts are different. Intuitively, it seems reasonable to penalize the similarity score based on this semantic discrepancy.

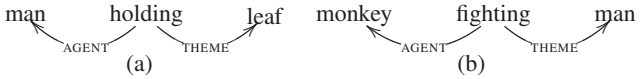

Figure 1: Semantic representation of 1(a) *A man is holding a leaf* and 1(b) *A monkey is fighting a man*.

This paper presents a novel approach to predict textual similarity. The main contributions are: (1) semantic representations are incorporated; (2) three logic form transformations capturing different levels of knowledge are used; and (3) semantic features are derived from a logic prover.

2 Related Work

Predicting similarity between text snippets is relevant to information retrieval, paraphrase recognition, grading answers to questions and many others. In this section, we focus on recent work and emphasize the differences with our approach.

The SemEval 2012 Task 6: A Pilot on Semantic Textual Similarity (Agirre et al. 2012) brought together 35 teams that competed against each other. The top 3 performers (Bär et al. 2012; Šarić et al. 2012; Banea et al. 2012) followed a machine learning approach with features that do not take into account the semantic structure of sentences, e.g., n-grams, word overlap, pairwise word similarity (corpus and knowledge based), dependency parses. Top-performers also used knowledge derived from large corpora, e.g., Wikipedia.

Participants that incorporated information about the structure of sentences (Glinos 2012; AbdelRahman and Blake 2012; Rios, Aziz, and Specia 2012) performed worse than the above systems. Out of 88 runs, they ranked 23, 36 and 64. We believe this is because they used semantic roles to create a "role-based similarity score" or did not combine deeper semantic information with more basic approaches that for certain sentence pairs yield very good results.

3 Approach

The main components of our approach are summarized in Figure 2. First, sentences are transformed into logic forms (lft_1, lft_2). Then, a modified logic prover is used to find a proof in both directions (lft_1 to lft_2, and lft_2 to lft_1). The prover yields similarity scores based on how many predicates had to be dropped and features characterizing the

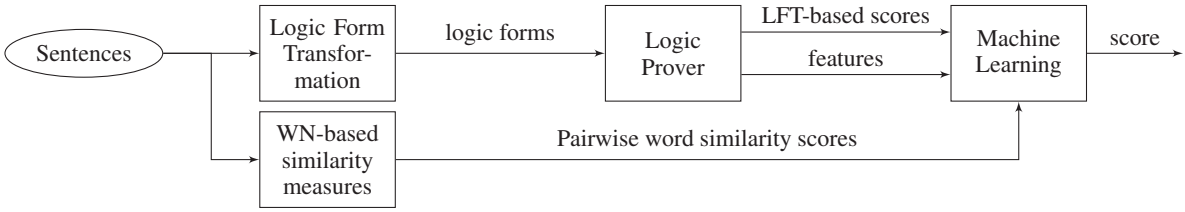

Figure 2: Main components of our semantic similarity prediction system.

	sentence: *A woman dances in the rain outside.*
	semantic relations extracted: AGENT(*dances*, *woman*), LOCATION(*dances*, *rain*)
Basic	`woman_N(`x_1`) & dance_V(`x_2`) & rain_N(`x_3`) & outside_M(`x_4`)`
SemRels	`woman_N(`x_1`) & dance_V(`x_2`) & AGENT_SR(`x_2, x_1`) & rain_N(`x_3`) & LOCATION_SR(`x_2, x_3`)`
Full	`woman_N(`x_1`) & dance_V(`x_2`) & AGENT_SR(`x_2, x_1`) & rain_N(`x_3`) & LOCATION_SR(`x_2, x_3`) & outside_M(`x_4`)`

Table 1: Examples of logic form transformations using *Basic*, *SemRels* and *Full* modes.

proofs. Additional similarity scores are obtained using WN-based similarity measures. Finally, all scores and features are combined using machine learning to obtain the final similarity score. The rest of this section details each component and exemplifies it with the sentences *A woman is dancing in the rain* and *A woman dances in the rain outside*.

3.1 Logic Form Transformation

The logic form transformation (LFT) of a sentence is derived from the concepts in it, the semantic relations linking them and named entities. We distinguish six types of predicates: (1) `N` for nouns, e.g., *woman*: `woman_N(`x_1`)`; (2) `V` for verbs, e.g., *dances*: `dance_V(`x_2`)`; (3) `M` for adjectives and adverbs, e.g., *outside*: `outside_M(`x_3`)`; (4) `O` for concepts encoded by other POS tags; (5) `NE` for named entities, e.g., *guitar*: `guitar_N(`x_4`) & instrument_NE(`x_4`)`; and (6) `SR` for semantic relations, e.g., *A woman dances*: `woman_N(`x_1`) & dance_V(`x_2`) & AGENT_SR(`x_2, x_1`). AGENT_SR(`x_2, x_1`)` could be read "x_2 *has agent* x_1".

In order to overcome semantic relation extraction errors, we have experimented with three logic form transformation modes. Each mode captures different levels of knowledge:

Basic generates predicates for nouns, verbs, modifiers and named entities. This logic form is parallel to accounting for content words, their POS tags and named entity types.

SemRels generates predicates for all semantic relations, concepts that are arguments of relations and named entities. If no semantic relations are found, this mode backs off to *Basic* to avoid generating an empty logic form.

Full generates predicates for all concepts, all semantic relations and all named entities.

Table 1 exemplifies the three logic forms. If perfect semantic relations were always available, *SemRels* would be the preferred mode. However, this is often not the case and combining the three yields better performance. Note that there is no predicate for *outside* in *SemRels* since it is not an argument of a semantic relation extracted.

3.2 Modified Logic Prover

Textual similarity is symmetric and therefore we find proofs in both directions (lft_1 to lft_2, and lft_2 to lft_1). In the rest of this section we only exemplify one, lft_1 to lft_2. The logic prover is a modification of OTTER (McCune and Wos 1997). For the textual similarity task, we load lft_1 and $\neg lft_2$ to the *set of support* and do not load anything to the *usable list*. Then, the logic prover begins its search for a proof. Two scenarios are possible: (1) a contradiction is found, i.e., a proof is found, and (2) a contradiction cannot be found. The modifications to the standard resolution procedure are used in scenario (2). In this case, predicates from lft_1 are dropped until a proof is found. The goal is to force the prover to always find a proof, and penalize partial proofs accordingly.

Predicate Dropping Criteria. Individual predicates from lft_1 are dropped following a greedy criterion: predicates that occur earlier are dropped first. After dropping a predicate, predicates that become unbound are dropped as well. Specifically, dropping a noun, verb or modifier may make a semantic relation or named entity predicate unbound. To avoid predicting high similarity between sentences with a *common* semantic structure but *unrelated* concepts instantiating this structure, we drop predicates for semantic relations and named entities when they become unbound.

Proof Scoring Criterion. The score of the proof from lft_1 to lft_2 is the ratio of predicates not dropped to find the proof over the number of predicates in lft_1. Note that the dropping mechanism, and in particular whether predicates that become unbound are automatically dropped, greatly impacts the proof obtained and its score. Table 2 exemplifies the dropping and scoring criteria. If unbound predicates were not dropped, the overall score would be $2/8 = 0.25$.

Feature Selection. While the proof score can be used directly as an estimator of the similarity between lft_1 and lft_2, we also extract features from the proof itself. Namely, for each predicate type (`N`, `V`, `M`, `O`, `SR`, `NE`), we count the number of predicates present in lft_1, the number of predicates dropped to find a proof for lft_2 and the ratio of the two counts. An example is provided in Table 3.

sent$_1$: *A woman plays an electric guitar*	sent$_2$: *A man is cutting a potato*
lft_1: woman_N(x_1) & play_V(x_2) & AGENT_SR(x_2, x_1) & electric_M(x_3) & guitar_N(x_4) & instrument_NE(x_4) & VALUE_SR(x_4, x_3) & THEME_SR(x_2, x_4)	
$\neg lft_2$: $\neg$man_N(x_1) $\vee$ $\neg$cut_V(x_2) $\vee$ $\neg$AGENT_SR(x_2, x_1) $\vee$ $\neg$potato_N(x_3) $\vee$ $\neg$THEME_SR(x_2, x_3)	

Step	Predicate dropped (regular)	Score	Predicate dropped (unbound)	Score
1	woman_N(x_1)	0.875	n/a	0.875
2	play_V(x_2)	0.750	AGENT_SR(x_2, x_1)	0.625
3	electric_M(x_3)	0.500	n/a	0.500
4	guitar_N(x_4)	0.375	instrument_NE(x_4), VALUE_SR(x_4, x_3), THEME_SR(x_2, x_4)	0.000

Table 2: Example of predicate dropping and proof scoring.

lft$_1$: woman_N(x_1) & dance_V(x_2) & AGENT_SR(x_2, x_1) & rain_N(x_3) & LOCATION_SR(x_2, x_3)																
lft$_2$: woman_N(x_1) & dance_V(x_2) & AGENT_SR(x_2, x_1) & rain_N(x_3) & LOCATION_SR(x_2, x_3) & outside_M(x_4)																
lft$_2$ to lft$_1$	pred. dropped	outside_M(x_4)														
	score	$5/6 = 0.8333$														
	features	n_t	n_d	n_r	v_t	v_d	v_r	m_t	m_d	m_r	ne_t	ne_d	ne_r	sr_t	sr_d	sr_r
		2	0	0	1	0	0	1	1	1	0	0	0	2	0	0

Table 3: Score and features obtained with the logic prover. For each predicate type (n, v, m, o, ne, sr; o omitted), features indicate the total number of predicates, the number of predicates dropped and ratio ($_t$, $_d$ and $_r$) until a proof is found.

Split	Score	Sentence Pair	Notes
MSRpar (36)	2.600	The unions also staged a five-day strike in March that forced all but one of Yale's dining halls to close. The unions also staged a five-day strike in March; strikes have preceded eight of the last 10 contracts.	Long sentences, difficult to parse; often several details are missing in one sentence but the pair is similar
MSRvid (13)	0.000	A woman is swimming underwater. A man is slicing some carrots.	Short sentences, easy to parse
SMTeuroparl (56)	4.250	Then perhaps we could have avoided a catastrophe. We would perhaps then able prevent a disaster.	One sentence often ungrammatical (SMT output), long sentences

Table 4: Examples from the three splits.The number between parenthesis indicates the average number of tokens per pair.

The LFT-based scores and features are fed to a machine learning algorithm. Specifically, we account for 9 scores (3×3; three scores (2 directions and average), three LFT modes) and 108 features ($3 \times 6 \times 3 \times 2 = 108$; three features for each of the six predicate types, three LFT modes, two directions).

3.3 Pairwise Word Similarity Measures and ML

Pairwise word similarity measures between concepts have been used for predicting textual similarity before. Following the proposal by Mihalcea, Corley, and Strapparava (2006), we derive scores using 7 measures: Path, LCH, Lesk, WUP, Resnik, Lin and JCN. We incorporate these scores for comparison purposes and to improve robustness in our approach.

We follow a supervised machine learning approach, where a model created with training instances is tested against unseen test instances. As a learning algorithm, we use bagging with M5P trees (Quinlan 1992; Wang and Witten 1997) as implemented in Weka (Hall et al. 2009).

4 Experiments and results

Logic forms are derived from the output of state-of-the-art NLP tools developed previously and not tuned in any way to the current task or corpora. Specifically, the named entity recognizer extracts up to 90 fine-grained types and was first develop for a Question Answering system (Moldovan et al. 2002). Semantic relations are extracted using a state-of-the-art semantic parser (Moldovan and Blanco 2012).

4.1 Corpora

We use the corpora released by the organizers of SemEval 2012 Task 06: A Pilot on Semantic Textual Similarity (Agirre et al. 2012). These corpora consist of pairs of sentences labeled with their semantic similarity score, ranging from 0.0 to 5.0. Sentence pairs come from three sources: (1) MSRpar, a corpus of paraphrases; (2) MSRvid, short video descriptions; and (3) SMTeuroparl, output of machine translation systems and reference translations. Table 4 shows examples of the three sources, for more details about the corpora refer to the aforementioned citation.

4.2 Results and Error Analysis

Results obtained with out approach and the top-performer at SemEval-2012 Task 6 (Bär et al. 2012) are shown in Table 5. The three individual *LFT scores* are the scores obtained by the logic prover (average of both directions) with the corresponding logic form. *LFT scores + features* is a system combining 117 features (9 scores and 108 additional features derived from the proof, Section 3.2). *WN scores* is a system combining the 7 WN-based word similarity scores. Finally, *All* is a system combining all scores and features available.

Regarding LFT modes, *Basic* performs better than *SemRels* and *Full* performs better than *SemRels*. The only exception is with sentences from *SMTeuroparl*, where *SemRels* outperforms *Basic* by a negligible difference, $0.4728 - 0.4695 = 0.0033$. These results lead to the conclusions that

			Correlation
MSRpar	LFT score	Basic	0.5240
		SemRels	0.4318
		Full	0.5074
	LFT scores + features		0.5522
	WN scores		0.5052
	All		0.5852
	(Bär et al. 2012)		0.6830
MSRvid	LFT score	Basic	0.7295
		SemRels	0.6459
		Full	0.6665
	LFT scores + features		0.7716
	WN scores		0.8504
	All		0.8602
	(Bär et al. 2012)		0.8739
SMTeuroparl	LFT score	Basic	0.4695
		SemRels	0.4728
		Full	0.4978
	LFT scores + features		0.4724
	WN scores		0.5111
	All		0.5180
	(Bär et al. 2012)		0.5280

Table 5: Correlations obtained for the test split by our approach and the top-performer at SemEval-2012 Task 6.

including concepts that are not arguments of relations (Full) is better than disregarding them (SemRels), and that building a system grounded exclusively on semantic relations (SemRels) performs worse than simpler approaches that only account for concepts (Basic).

Combining LFT scores and features derived from the proofs, the main novelty of our approach, brings substantial improvements with MSRpar and MSRvid, but worse performance with SMTeuroparl. The fact that most pairs in SMTeuroparl include one ungrammatical sentence makes our NLP tools perform poorly, greatly affecting overall performance. We note, though, that when sentences are easier to parse (MSRpar, MSRvid), the benefits are clear.

Only using scores from WN-based word similarity measures performs astonishingly well. *WN scores* outperforms *LFT scores + features* except in MSRpar. We believe that this is due to the fact that sentences in MSRvid are very short (13 words on average per pair), and the grammar issue in SMTeuroparl pointed out above.

Finally, best results are obtained when all scores and features are combined. This suggests that while WN-based scores provide a strong baseline, it can be improved by incorporating features capturing the semantic structure of sentences. Also, semantic information brings improvements only when combined with simpler methods, as the results obtained by *All*, *LFT scores + features* and *LFT score* with the three LFT modes show.

Comparison with Previous Work Our best system, *All*, performs worse than the top performer (Table 5): -0.0978 (MSRpar), -0.0137 (MSRvid) and -0.100 (SMTeuroparl). We note, though, that (1) the differences are small for MSRvid and SMTeuroparl, and (2) our approach does not require knowledge from Wikipedia or other large corpora.

References

AbdelRahman, S., and Blake, C. 2012. Sbdlrhmn: A rule-based human interpretation system for semantic textual similarity task. In *Proceedings of SemEval 2012*, 536–542.

Agirre, E.; Cer, D.; Diab, M.; and Gonzalez-Agirre, A. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of SemEval 2012*, 385–393.

Banea, C.; Hassan, S.; Mohler, M.; and Mihalcea, R. 2012. Unt: A supervised synergistic approach to semantic text similarity. In *Proceedings of SemEval 2012*, 635–642.

Bär, D.; Biemann, C.; Gurevych, I.; and Zesch, T. 2012. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In *Proceedings of SemEval 2012*, 435–440.

Dolan, W. B., and Brockett, C. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Giampiccolo, D.; Magnini, B.; Dagan, I.; and Dolan, B. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, 1–9.

Glinos, D. 2012. Ata-sem: Chunk-based determination of semantic text similarity. In *Proceedings of SemEval 2012*, 547–551.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The weka data mining software: an update. *SIGKDD Explor. Newsl.* 11(1):10–18.

McCune, W., and Wos, L. 1997. Otter: The cade-13 competition incarnations. *Journal of Automated Reasoning* 18:211–220.

Mihalcea, R.; Corley, C.; and Strapparava, C. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st national conference on Artificial intelligence*, AAAI'06, 775–780.

Moldovan, D., and Blanco, E. 2012. Polaris: Lymba's semantic parser. In *Proceedings of the Eight Int. Conference on Language Resources and Evaluation (LREC'12)*.

Moldovan, D.; Harabagiu, S.; Girju, R.; Morarescu, P.; Lacatusu, F.; Novischi, A.; Badulescu, A.; and Bolohan, O. 2002. Lcc tools for question answering. In Voorhees, and Buckland., eds., *Proceedings of the 11th Text REtrieval Conference (TREC-2002)*.

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

Rios, M.; Aziz, W.; and Specia, L. 2012. Uow: Semantically informed text similarity. In *Proceedings of SemEval 2012*, 673–678.

Šarić, F.; Glavaš, G.; Karan, M.; Šnajder, J.; and Dalbelo Bašić, B. 2012. Takelab: Systems for measuring semantic text similarity. In *Proceedings of SemEval 2012*, 441–448.

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