A Semantically Enhanced Approach to Determine Textual Similarity

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

Abstract

This paper presents a novel approach to determine textual similarity. A layered methodology to transform text into logic forms is proposed, and semantic features are derived from a logic prover. Experimental results show that incorporating the semantic structure of sentences is beneficial. When training data is unavailable, scores obtained from the logic prover in an unsupervised manner outperform supervised methods.

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) below is very similar [5 out of 5], (2) is somewhat similar [3 out of 5] and (3) is not similar at all [0 out of 5]:

1. Someone is removing the scales from the fish.
   A person is descaling a fish.
2. A woman is chopping an herb.
   A man is finely chopping a green substance.
3. A cat is playing with a watermelon on a floor.
   A man is pouring oil into a pan.

State-of-the-art systems to determine textual similarity (Bär et al., 2012; Šarić et al., 2012; Banea et al., 2012) do not account for the semantic structure of sentences, and mostly rely on word pairings and knowledge derived from large corpora, e.g., Wikipedia. Regardless of details, each word in sent$_1$ is paired with the word in sent$_2$ that is most similar according to some similarity measure. Then, all similarities are added and normalized by the length of sent$_1$ to obtain the similarity score from sent$_1$ to sent$_2$. The process is repeated to obtain the similarity score from sent$_2$ to sent$_1$, and both scores are then averaged to determine the overall textual similarity. Several word-to-word similarity measures are often combined with other shallow features, e.g., n-gram overlap, syntactic dependencies, to obtain the final similarity score.

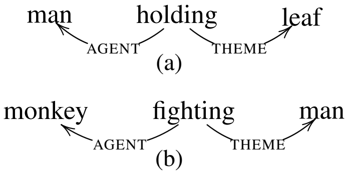

Figure 1: Semantic representation of 1(a) *A man is holding a leaf* and 1(b) *A monkey is fighting a man.*

Consider sentences 1(a) *A man is holding a leaf* and 1(b) *A monkey is fighting 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 word '*man*' in 1(a) will find '*man*' from 1(b) to be a perfect match. One of three content words is a match and thus the estimated 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*' plays the role of AGENT in 1(a), and THEME in 1(b). While in both sentences the word '*man*' encodes the same concept, their semantic functions with respect to other concepts are different. Intuitively, it seems reasonable to penalize the similarity score based on the role discrepancy.

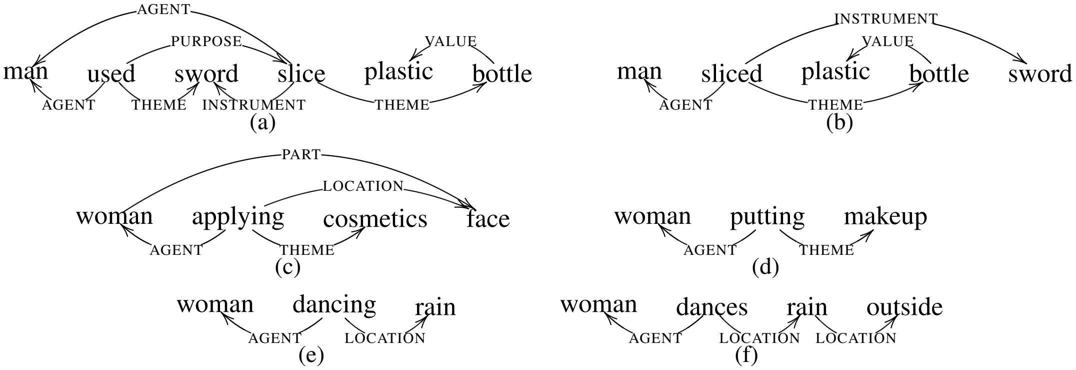

Figure 2: Semantic representations of 2(a) *The man used a sword to slice a plastic bottle*, 2(b) *A man sliced a plastic bottle with a sword*, 2(c) *A woman is applying cosmetics to her face*, 2(d) *A woman is putting on makeup*, 2(e) *A woman is dancing in the rain*, and 2(f) *A woman dances in the rain outside*. Pairs (a, b), (c, d) and (e, f) are highly similar even though concepts and relations only match partially.

This paper proposes a novel approach to determine textual similarity. Semantic representations of sentences are exploited, syntactic features omitted and the only external resource used in WordNet (Miller, 1995). The main novelties of our approach are: it (1) derives semantic features from a logic prover to be used in a machine learning framework; (2) uses three logic form transformations capturing different levels of knowledge; and (3) incorporates semantic representations extracted automatically.

1.1 Matching Semantic Representations and Determining Textual Similarity

Throughout this paper, the semantic representation of a sentence comprises the concepts in it, semantic relations linking those concepts and named entities qualifying them. First, we note that existing tools to extract semantic relations and named entities are not perfect, thus any system relying on them will suffer from incomplete and incorrect representations. Second, even if flawless representations were readily available, the problem of determining textual similarity cannot be reduced to matching semantic representations: partial matches may correspond to completely similar sentences. The rest of this section illustrates this point with the examples in Figure 2. Our approach (Section 3) copes with the inherent errors made by tools used to obtain semantic representations and learns which parts of a representation are important to determine textual similarity.

Consider sentences 2(a) *The man used a sword to slice a plastic bottle* and 2(b) *A man sliced a plastic bottle with a sword*. Both sentences have high similarity [5 out of 5], and yet their semantic representations only match partially. In this example, the verb '*used*' in 2(a) and its semantic links are somewhat semantically superfluous. Note that in other cases, missing a semantic relation signals lower similarity, e.g., *I had fun [at the party]*$_{\text{LOCATION}}$ and *I had fun*, while similar, do not convey the same meaning.

Sentence 2(c) *A woman is applying cosmetics to her face* and 2(d) *A woman is putting on makeup* are highly similar even though the latter specifies neither the LOCATION where the '*makeup*' is applied nor the fact that a PART of the '*woman*' is her '*face*'. Similarly, sentences 2(e) *A woman is dancing in the rain* and 2(f) *A woman dances in the rain outside* are semantically equivalent since '*rain*' always has LOCATION '*outside*': missing this information does not carry loss of meaning.

2 Related Work

Determining similarity between text snippets is relevant to information retrieval (Hatzivassiloglou et al., 1999), paraphrase recognition (Madnani and Dorr, 2010), grading answers to questions (Mohler et al., 2011) and many others. We focus on recent work and emphasize the differences from 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 ma-

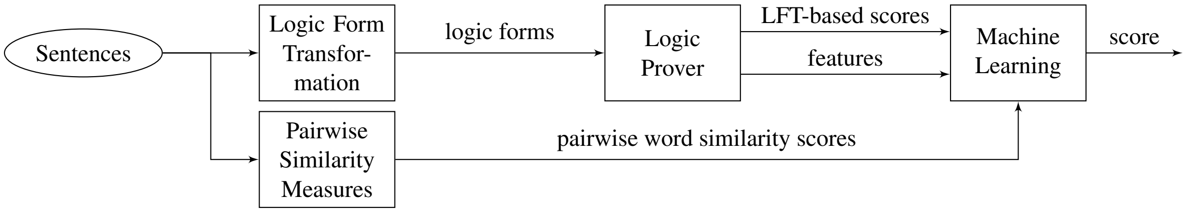
Sentences
Logic Form Transformation
logic forms
Logic Prover
LFT-based scores
features
Machine Learning
score
Pairwise Similarity Measures
pairwise word similarity scores

Figure 3: Main components of our system to determine textual similarity.

chine learning approach with features that do not take into account the semantic structure of sentences, e.g., n-grams, word overlap, evaluation measures for machine translation, pairwise word similarities, syntactic dependencies. All three used WordNet, Wikipedia and other large corpora. In particular, Banea et al. (2012) obtained models from 6 million Wikipedia articles and more than 9.5 million hyperlinks; Bär et al. (2012) used Wiktionary[1], which contains over 3 million entries; and Šarić et al. (2012) used The New York Times Annotated Corpus (Sandhaus, 2008), which contains over 1.8 million news articles, and Google n-grams (Lin et al., 2012), which consists of approximately 24GB of compressed text files. Our approach only uses WordNet, by far the smallest external resource with less than 120,000 synsets.

Participants that incorporated information about the semantic structure of sentences (Glinos, 2012; Rios et al., 2012)[2] did not perform at the top. Out of 88 runs, they were ranked 16, 36 and 64. We believe this is because they use semantic relations to calculate some ad-hoc similarity score. In contrast, our approach derives features from semantic representations encoded using logic, and combine these features using machine learning. Moreover, we use three logic form transformations capturing different levels of knowledge, from only content words to semantic structure. In turn, this allows us to boost performance by relying on semantics when simpler shallow methods fail.

A few logic-based approaches to recognize textual entailment are similar to the work presented here. Bos and Markert (2006) extract semantic representations with Boxer (Bos et al., 2004) and incorporate background knowledge from external resources. They use a standard theorem prover and extract 8 features that are later combined using machine learning. Raina et al. (2005) use a logic form transformation derived from dependency parses and named entities. They use abductive reasoning and define an assumption cost model to account for partial entailments. Unlike them, we define three logic from transformations, use a modified resolution step and extract hundreds of features from the proofs. Tatu and Moldovan (2005) use a modified logic prover that drops predicates when a proof cannot be found. Unlike us, they do not drop unbound predicates and use a single logic form transformation. Another key difference is that they assign fixed weights to predicates a priori instead of using machine learning to determine them.

3 Approach

Our approach to determine textual similarity (Figure 3) is grounded on using semantic features derived from a logic prover that are later combined in a standard supervised machine learning framework. 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 the number of predicates dropped and features characterizing the proofs. Additional similarity scores are obtained using standard pairwise word similarity measures. Finally, all scores and features are combined using machine learning to yield the final textual similarity score.

If training data is unavailable, only the LFT-based and individual pairwise word similarity scores apply, the machine learning component is the only one supervised. The rest of this section details each component and exemplifies it with 2(e) *A woman is dancing in the rain* and 2(f) *A woman dances in the rain outside*.

[1] http://www.wiktionary.org/

[2] A third team, *spirin2*, submitted results but a description paper could not be found in the ACL anthology.

	sent$_1$: *A woman is dancing in the rain.*
	semantic relations extracted: AGENT(*dancing*, *woman*), LOCATION(*dancing*, *rain*)
Basic	woman_N(x_1) & dance_V(x_2) & rain_N(x_3)
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)
	sent$_2$: *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 from transformation using modes *Basic*, *SemRels* and *Full*.

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. Unlike other LFT proposals (Zettlemoyer and Collins, 2005; Poon and Domingos, 2009), transforming sentences into logic forms is a straightforward step, the quality of the logic forms is determined by the output of standard NLP tools.

We distinguish six types of predicates:

- N for nouns, e.g., *woman*: woman_N(x_1).
- V for verbs, e.g., *dances*: dance_V(x_2).
- M for adjectives and adverbs, e.g., *outside*: outside_M(x_3).
- O for concepts encoded by other POS tags.
- NE for named entities, e.g., *guitar*: guitar_N(x_4) & instrument_NE(x_4).
- SR for semantic relations, e.g., *A woman dances*: woman_N(x_1) & dance_V(x_2) & AGENT_SR(x_2,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 all 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 qualifying those concepts. This mode ignores concepts not linked to other concepts through a relation and might miss key concepts if some relations are missing. If no semantic relations are found, this mode backs off to *Basic* to avoid empty logic forms.

Full generates predicates for all concepts, all semantic relations and all named entities. It is equivalent to *SemRels* after adding predicates for concepts that are not arguments of a semantic relation.

Table 1 exemplifies the three logic form modes. If perfect semantic relations were always available, *SemRels* would be the preferred mode. However, this is often not the case and combining the three logic forms yields better performance (Section 4). Note that since relation LOCATION(*rain*, *outside*) is not extracted from sent$_2$, predicate outside_M(x_4) is not present in mode *SemRels*.

3.2 Modified Logic Prover

Textual similarity is symmetric and therefore we find proofs in both directions (from *lft*$_1$ to *lft*$_2$ and from *lft*$_2$ to *lft*$_1$). The logic prover uses a modified resolution procedure to calculate a similarity score and features derived from the proof. The rest of this section exemplifies one direction, *lft*$_1$ to *lft*$_2$. The logic prover is a modification of OTTER[3] (McCune and Wos, 1997), an automated theorem prover for first-order logic. For the textual similarity task, we load *lft*$_1$ and $\neg$*lft*$_2$ to the *set of support* and lexical chain axioms 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; or (2) a contradiction cannot

[3] http://www.cs.unm.edu/~mccune/otter/

<table>
<tr><td colspan="3">sent$_1$: <i>A man is cutting a potato</i></td><td colspan="2">sent$_2$: <i>A woman plays an electric guitar</i></td></tr>
<tr><td colspan="5">lft_1: man_N(x_1) & cut_V(x_2) & AGENT_SR(x_2, x_1) & potato_N(x_3) & THEME_SR(x_2, x_3)</td></tr>
<tr><td colspan="5">$\neg lft_2$: $\neg$woman_N(x_1) $\vee$ $\neg$play_V(x_2) $\vee$ $\neg$AGENT_SR(x_2, x_1) $\vee$ $\neg$electric_M(x_3) $\vee$ $\neg$guitar_N(x_4) $\vee$ $\neg$instrument_NE(x_4) $\vee$ $\neg$VALUE_SR(x_4, x_3) $\vee$ $\neg$THEME_SR(x_2, x_4)</td></tr>
<tr><td>Step</td><td>Predicate dropped (regular)</td><td>Score</td><td>Predicate dropped (unbound)</td><td>Score</td></tr>
<tr><td>1</td><td>woman_N(x_1)</td><td>0.875</td><td>n/a</td><td>0.875</td></tr>
<tr><td>2</td><td>play_V(x_2)</td><td>0.750</td><td>AGENT_SR(x_2, x_1)</td><td>0.625</td></tr>
<tr><td>3</td><td>electric_M(x_3)</td><td>0.500</td><td>n/a</td><td>0.500</td></tr>
<tr><td>4</td><td>guitar_N(x_4)</td><td>0.375</td><td>instrument_NE(x_4), VALUE_SR(x_4, x_3), THEME_SR(x_2, x_4)</td><td>0.000</td></tr>
</table>

Table 2: Example of predicate dropping step by step. Predicates AGENT_SR(x_2, x_1) and THEME_SR(x_2, x_4) would not be dropped if unbound predicates were not dropped, yielding a score of 0.250 instead of 0.000.

be found. The modifications to the standard resolution procedure are used in scenario (2), when a proof cannot be found. In this case, predicates from $\neg lft_2$ are dropped until a proof is found. The worst case occurs when all predicates in $\neg lft_2$ are dropped. The goal of the dropping mechanism is to force the prover to always find a proof, and penalize partial proofs accordingly.

Lexical chain axioms are extracted from WordNet. Assuming each word w in sent$_1$ has the first sense, axioms $w \rightarrow c$, where c is at most distance 2 in the WordNet hierarchy are generated. For example, axioms derived from *woman* include *woman* → *female*, *woman* → *mistress*, *woman* → *widow* and *woman* → *madam*. Although simple, this WordNet expansion proved useful in our experiments.

3.2.1 Predicate Dropping Criteria

When a proof cannot be found, individual predicates from $\neg lft_2$ not unifiable with predicates in lft_1 are dropped. A greedy algorithm was implemented for this step: out of all eligible predicates, drop whichever occurs first.

Dropping a predicate is not done in isolation. After dropping a predicate, all predicates that become unbound are dropped as well. With our current logic form transformation, dropping a noun, verb or modifier may make a semantic relation (_SR) or named entity (_NE) predicate unbound. To avoid determining high similarity between sentences with a *common* semantic structure but *unrelated* concepts instantiating this structure, predicates encoding semantic relations and named entities are automatically dropped when they become unbound.

3.2.2 Proof Scoring Criterion

The score assigned to the proof from lft_1 to lft_2 is calculated as the ratio of number of predicates in $\neg lft_2$ not dropped to find the proof over the original number of predicates in $\neg lft_2$.

Note that the dropping mechanism, and in particular whether predicates that become unbound are automatically dropped, greatly impact the proof obtained and its score (Table 2). If predicates that become unbound were not automatically dropped in each step, instrument_NE(x_4) and VALUE_SR(x_4, x_3) would be dropped in steps 5 and 6, AGENT_SR(x_2, x_1) and THEME_SR(x_2, x_4) would not be dropped, and the final score would be 0.250 instead of 0.000. In plain English, dropping unbound predicates avoids matching semantic structures instantiated by unrelated concepts.

3.2.3 Feature Selection

While the proof score can be used directly as an estimator of the similarity between lft_1 and lft_2, additional features are extracted from the proof itself. Namely, for each predicate type (N, V, M, O, SR, NE), we count the number of predicates in lft_2, the number of predicates dropped to find a proof and the ratio of the two counts. These three counts are also calculated for each specific semantic relation predicate (AGENT_SR, LOCATION_SR, etc.). An example of score and feature calculation in both directions is shown in Table 3.

The LFT-based scores and features are fed to a machine learning algorithm. Specifically, there are 477 features derived from the logic prover:

- 9 LFT-based scores (3×3; three scores (2 directions and average), three LFT modes)

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$_1$ to lft$_2$	pred. dropped	outside_M(x_4)														
	score	$5/6 = 0.833$														
	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
lft$_2$ to lft$_1$	pred. dropped	none														
	score	1														
	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	0	0	0	0	0	0	2	0	0

Table 3: Two logic forms and output of logic prover in both directions. For each predicate type (n, v, m, o, ne, sr) and semantic relation type (AGENT, LOCATION, etc.) features indicate the total number of predicates, the number of predicates dropped until a proof is found and ratio of the two counts ($_t$, $_d$ and $_r$ respectively). We omit the features for predicate O and individual semantic relations because of space constraints.

- 108 features for predicates ($3 \times 6 \times 3 \times 2 = 108$; three features for each of the six predicate types, three LFT modes, two directions)
- 360 features specific to a semantic relation ($3 \times 20 \times 3 \times 2 = 360$; three features for each of the 20 semantic relations types, three LFT modes, two directions)

3.3 Pairwise Word Similarities

Pairwise word similarity measures between concepts have been long studied, and they have been used for the task of textual similarity before (Mihalcea et al., 2006). We incorporate scores derived using these measures for comparison purposes and to improve robustness in our approach.

Basically, each open-class word in $sent_1$ is paired with the open-class word in $sent_2$ that is most similar according to some similarity measure. All these individual similarities are summed and normalized by the length of $sent_1$ to find the similarity between $sent_1$ and $sent_2$. The process is repeated from $sent_2$ to $sent_1$ to obtain the similarity between $sent_2$ and $sent_1$, and both overall similarities are averaged to determine the final similarity score.

We have experimented with measures Path (distance in a taxonomy), LCH (Leacock and Chodorow, 1998), Lesk (Lesk, 1986), WUP (Wu and Palmer, 1994), Resnik (Resnik, 1995), Lin (Lin, 1998) and JCN (Jiang and Conrath, 1997), and use the WordNet::Similarity package[4].

3.4 Machine Learning Algorithm

We follow a standard supervised machine learning framework. Instances from the training split are used to create a model that is later tested with test instances not seen during training. The model was tuned using 10-fold cross-validation over the training instances. As a learning algorithm, we use bagging with M5P decision trees (Quinlan, 1992; Wang and Witten, 1997) as implemented in the Weka software package (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. Our approach is not tied to any tool, set of named entities or relations. Any other semantic representation could be used; the only required modification would be the LFT component (Figure 3) so that it accounts for the subtleties of the representation of choice.

The named entity recognizer extracts 35 fine-grained types organized in a taxonomy (date, language, city, instrument, etc.) and was first developed for a question answering system (Moldovan et al., 2002). The implementation uses publicly available gazetteers as well as machine learning.

Semantic relations are extracted with Polaris (Moldovan and Blanco, 2012), a semantic parser that given text extracts semantic relations. Polaris is trained using FrameNet (Baker et al., 1998), PropBank (Palmer et al., 2005), NomBank (Meyers et al., 2004), several SemEval corpora (Girju et al., 2007;

[4] http://wn-similarity.sourceforge.net/

	Score	Sentence Pair	Notes
MSRpar (36/35) [750/750]	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 (13/13), [750/750]	0.000	A woman is swimming underwater. A man is slicing some carrots.	Short sentences, easy to parse
SMTeuroparl (56/21), [734/459]	4.250	Then perhaps we could have avoided a catastrophe. We would perhaps then able prevent a disaster.	One sentence often ungrammatical (SMT)
surprise.OnWN (–/16), [–/750]	1.500	the alleviation of distress a change for the better.	WN glosses, difficult to parse with standard tools
surprise.SMTnews (–/24), [–/399]	3.000	He did, but the initiative did not get very far What he has done without the initiative goes too far.	One sentence often ungrammatical (SMT)

Table 4: Examples of sentence pairs belonging to the five sources. The numbers between round (square) parenthesis indicate the average number of tokens per sentence pair (number of instances) in the train and test splits.

Pustejovsky and Verhagen, 2009; Hendrickx et al., 2010) and in-house annotations.

4.1 Corpora

We use the corpora released by SemEval 2012 Task 06: A Pilot on Semantic Textual Similarity[5] (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 five sources: (1) MSRpar, a corpus of paraphrases; (2) MSRvid, short video descriptions; (3) SMTeuroparl, output of machine translation systems and reference translations; (4) surprise.OnWN, OntoNotes (Hovy et al., 2006) and WordNet (Miller, 1995) glosses; and (5) surprise.SMTnews, output of machine translation systems in the news domain and gold translations. Examples can be found in Table 4, for more details refer to the aforementioned citation.

4.2 Results and Error Analysis

Results are reported using the same train and test splits provided by the organization of SemEval 2012 Task 6. For surprise.OnWn and surprise.SMTnews, only test data is available and supervised machine learning is not an option.

Table 5 shows results obtained with the test split not dropping and dropping unbound predicates. For comparison purposes, results of the top-3 performers and participants using the semantic structure of sentences are also shown. *LFT-score* systems output the score (average of both directions) obtained with the corresponding logic form transformation (Basic, SemRels or Full) and are unsupervised: training data with textual similarity scores is not used. The other three systems presented are supervised. *LFT-scores + features* combines the 9 LFT-scores and 468 features derived from the logic proof. *WN-scores* uses as features the 7 scores derived using pairwise word similarity measures. Finally, *All* combines the full set of 484 features. We indicate that the performance of one of our systems with respect to *LFT score Basic not dropping unbound predicates* is significant with * (confidence 99%) and † (confidence 95%).

Overall, systems that drop unbound predicates perform better than systems that do not drop them. The only noticeable exception is *LFT-score* with sentences from SMTeuroparl. However, best results for SMTeuroparl are obtained dropping unbound predicates and using *All* features. Henceforth, we comment on results dropping unbound predicates as they are higher.

Regarding logic form transformations, one can see a trend depending on the source of sentences. Polaris, the semantic parser, and the syntactic parser Polaris relies on are mostly trained in the news domain, and thus semantic representations have higher quality in that domain. For SMTeuroparl and SMTnews, the two corpora closest to the news domain, *Full* obtains better results than *Basic* and *SemRels*. The difference is most noticeable in SMTnews, where *Basic* yields 0.4616, *SemRels*

[5] http://www.cs.york.ac.uk/semeval-2012/task6/

	System			MSRpar	MSRvid	SMTeuroparl	OnWN	SMTnews
not dropping unbound predicates	noML	LFT score	Basic	0.4963	0.8198	0.5101	0.6103	0.4588
			SemRels	*0.3952	*0.6753	0.4920	*0.5055	0.4477
			Full	0.4525	*0.7024	0.5183	0.5895	0.4956
	ML	LFT scores + features		*0.5750	0.8466	0.4725	n/a	n/a
		WN scores		0.4978	0.8495	0.5217	n/a	n/a
		All		*0.5992	†0.8660	0.5194	n/a	n/a
dropping unbound predicates	noML	LFT score	Basic	†0.5552	0.8234	0.4994	0.6120	0.4616
			SemRels	0.4556	*0.7388	0.4871	*0.5113	0.4796
			Full	0.5250	*0.7672	0.5130	0.5895	†0.5291
	ML	LFT scores + features		*0.5770	0.8440	0.5277	n/a	n/a
		WN scores		0.4977	0.8495	0.5217	n/a	n/a
		All		***0.6157**	***0.8709**	†**0.5745**	n/a	n/a
Top performer	(Bär et al., 2012)			0.6830	0.8739	0.5280	0.6641	0.4937
	(Šarić et al., 2012)			0.6985	0.8620	0.3612	0.7049	0.4683
	(Banea et al., 2012)			0.5353	0.8750	0.4203	0.6715	0.4033
Team w/ semantic structure	spirin2			0.5769	0.8203	0.4667	0.5835	0.4945
	(Rios et al., 2012)			0.3628	0.6426	0.3074	0.2806	0.2082
	(Glinos, 2012)			0.2312	0.6595	0.1504	0.2735	0.1426

Table 5: Correlations obtained with the test split using our approach (not dropping and dropping unbound predicates), and results obtained by the top-3 performers and teams that included in their models features derived from the semantic structure of sentences. Statistically significant differences in performance between our systems and *LFT score Basic not dropping unbound predicates* are indicated with * (confidence 99%) and † (confidence 95%).

0.4796 (+0.0180) and *Full* 0.5291 (+0.0675 and +0.0495 respectively).

Outside the news domain (MSRpar, MSRvid, OnWN), *Basic* performs better than *SemRels* and *Full*, and *Full* performs better than *SemRels*. This leads to the conclusion that several semantic relations are often missing, and thus considering concepts even if they are not linked to other concepts via a semantic relation (*Full*) is more sound than ignoring them (*SemRels*).

When training data is available (MSRpar, MSRvid, SMTeuroparl), *LFT-scores + features* always outperforms the scores obtained with a single logic form transformation in an unsupervised manner. In other words, combining the scores obtained with the three logic form transformations and incorporating the additional features derived from the proofs improves performance. These results demonstrate that while a shallow logic form transformation (*Basic*) offers a strong baseline, it can be successfully complemented with logic form transformations that consider the semantic structure of sentences (*SemRels*, *Full*) and additional features characterizing the proofs. The improvements *LFT-scores + features* brings over the LFT-score obtained with *Basic* are substantial: 0.0218 (3.9%) for MSRpar, 0.0206 (2.5%) for MSRvid and 0.0283 (5.7%) for SMTeuroparl.

WN scores, which only uses as features the scores derived from pairwise word similarity measures, performs astonishingly well for some corpora. Namely, the differences in performance between *LFT scores + features* and *WN scores* in MSRvid and SMTeuroparl are minimal (−0.0055 and +0.0060). We believe this is due to the characteristics of these two corpora. Sentence pairs from MSRvid are very short with 13 tokens on average (Table 4), i.e., 6.5 tokens per sentence, and SMTeuroparl pairs are hard to parse: at least one comes from a machine translation system and is often ungrammatical.

Finally, dropping unbound predicates and using *All* features outperforms any other system. While both *LFT scores + features* and *WN scores* yield good performance, the combination of the two outperforms them. Features extracted successfully complement each other for all corpora.

4.2.1 A Look at the ML Model

A benefit of decision trees is that one can inspect them. This section briefly gives insight about the most predictive features for *All* system.

The best features, i.e., features used in decisions closer to the root, are the LFT-scores calculated using *Basic* and *Full*. The LFT-score obtained using *SemRels* is used only when the other two cannot discriminate. Sorted by impact, the features extracted for verbs, nouns, semantic relations, named entities and modifiers follow. Towards the bottom of the tree, features for specific semantic relations (AGENT_SR, LOCATION_SR, etc.) are used. All three sources (MSRpar, MSRvid and SMTeuroparl) use features for THEME, LOCATION, AGENT and QUANTIFICATION. MSRpar also benefits from features for TIME and only SMTeuroparl benefits from TOPIC and MANNER.

4.2.2 Comparison with Previous Work

The semantic logic-based approach presented in this paper either outperforms other systems or performs in the top-3 (Table 5). Moreover, it clearly outperforms any other proposal that takes into account the semantic structure of sentences. These results lead to the conclusion that the semantic structure of sentences is worth considering and more effort should be devoted to deeper approaches.

When using sentences in the news domain (SMTeuroparl and SMTnews), i.e., when text is closer to the domain in which the NLP tools are trained, our semantic approach yields the best results known to date. For MSRvid, the system presented here performs as well as systems that use external knowledge (Section 2), the differences are minimal (+0.0030, −0.0089, +0.0041) and not statistically significant (confidence 99%). For MSRpar, the system performs amongst the top-3 even though two of these systems clearly obtained better results (+0.0673, +0.0828); both differences are statistically significant (confidence 99%).

Performance using surprise.OnWN deserves special comment. This corpus contains definitions, not sentences (Table 4). Lin's similarity measure alone yields a correlation of 0.6787, beating all systems in Table 5 except one of the top-3 performers (Šarić et al., 2012). Our semantic approach is not successful because we cannot extract valid representations, glosses are rarely a full sentence and are hard to parse with generic NLP tools like the ones we use.

5 Conclusions

This paper presents a novel approach to determine textual similarity that employs a logic prover to extract semantic features. A layered methodology to transform text into logic forms using three logic form transformations modes is presented. Each mode captures different levels of knowledge, from only content words to semantic representations automatically extracted. Best results are obtained when features derived from the logic prover are complemented with simpler pairwise word similarity measures. Features that account for the semantic structure of sentences are incorporated when needed, as the results obtained with systems *All*, *LFT scores* and *WN scores* show.

Our approach is heavily dependent on the quality of semantic representations, and unlike current top performers, does not require knowledge derived from Wikipedia or other large corpora. State-of-the-art NLP tools to extract semantic representations from text, which are far from perfect, yield promising results. Indeed, the approach outperforms previous work when the source text is relatively familiar to the tools, i.e., within the news domain, and performs in the top-3 otherwise.

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 Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of the 17th international conference on Computational Linguistics*, Montreal, Canada.

Carmen Banea, Samer Hassan, Michael Mohler, and Rada Mihalcea. 2012. Unt: A supervised synergistic approach to semantic text similarity. In *Proceedings of the Sixth International Workshop on Se-*

mantic Evaluation (SemEval 2012), pages 635–642, Montréal, Canada, 7-8 June.

Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 2012. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 435–440, Montréal, Canada, 7-8 June.

Johan Bos and Katja Markert. 2006. Recognising textual entailment with robust logical inference. In *Proceedings of the First international conference on Machine Learning Challenges: evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, pages 404–426, Berlin, Heidelberg. Springer-Verlag.

Johan Bos, Stephen Clark, Mark Steedman, James R. Curran, and Julia Hockenmaier. 2004. Wide-coverage semantic representations from a ccg parser. In *Proceedings of Coling 2004*, pages 1240–1246, Geneva, Switzerland, Aug 23–Aug 27. COLING.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*. Association for Computational Linguistics.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague, June. Association for Computational Linguistics.

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, June. Association for Computational Linguistics.

Demetrios Glinos. 2012. Ata-sem: Chunk-based determination of semantic text similarity. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 547–551, Montréal, Canada, 7-8 June.

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

Vasileios Hatzivassiloglou, Judith L. Klavans, and Eleazar Eskin. 1999. Detecting text similarity over short passages: exploring linguistic feature combinations via machine learning. In *In Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 203–212.

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, July. Association for Computational Linguistics.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: the 90% Solution. In *NAACL '06: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers on XX*, pages 57–60, Morristown, NJ, USA. Association for Computational Linguistics.

J.J. Jiang and D.W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the Int'l. Conf. on Research in Computational Linguistics*.

C. Leacock and M. Chodorow, 1998. *Combining local context and WordNet similarity for word sense identification*, pages 305–332. In C. Fellbaum (Ed.), MIT Press.

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

Yuri Lin, Jean-Baptiste Michel, Erez Aiden Lieberman, Jon Orwant, Will Brockman, and Slav Petrov. 2012. Syntactic annotations for the google books ngram corpus. In *Proceedings of the ACL 2012 System Demonstrations*, pages 169–174, Jeju Island, Korea, July. Association for Computational Linguistics.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Nitin Madnani and Bonnie J. Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Comput. Linguist.*, 36(3):341–387, September.

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

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. Annotating noun argument structure for nombank. In *LREC*. European Language Resources Association.

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

George A. Miller. 1995. WordNet: A Lexical Database for English. In *Communications of the ACM*, volume 38, pages 39–41.

Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 752–762, Portland, Oregon, USA, June. Association for Computational Linguistics.

Dan Moldovan and Eduardo Blanco. 2012. Polaris: Lymba's semantic parser. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, and Jan Odijk a nd Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 66–72, Istanbul, Turkey, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1040.

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

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised Semantic Parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1–10, Singapore, August. Association for Computational Linguistics.

James Pustejovsky and Marc Verhagen. 2009. SemEval-2010 Task 13: Evaluating Events, Time Expressions, and Temporal Relations (TempEval-2). In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 112–116, Boulder, Colorado, June. Association for Computational Linguistics.

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

Rajat Raina, Andrew Y. Ng, and Christopher D. Manning. 2005. Robust textual inference via learning and abductive reasoning. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 3*, AAAI'05, pages 1099–1105. AAAI Press.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Miguel Rios, Wilker Aziz, and Lucia Specia. 2012. Uow: Semantically informed text similarity. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 673–678, Montréal, Canada, 7-8 June.

Evan Sandhaus. 2008. The new york times annotated corpus. In *Linguistic Data Consortium*, Philadelphia, PA.

Marta Tatu and Dan Moldovan. 2005. A semantic approach to recognizing textual entailment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 371–378, Stroudsburg, PA, USA. Association for Computational Linguistics.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. Takelab: Systems for measuring semantic text similarity. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 441–448, Montréal, Canada, 7-8 June.

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

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.

Luke Zettlemoyer and Michael Collins. 2005. Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars. In *Proceedings of the Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 658–666, Arlington, Virginia. AUAI Press.