UNITOR: Combining Semantic Text Similarity functions through SV Regression

Danilo Croce, Paolo Annesi, Valerio Storch and Roberto Basili
Department of Enterprise Engineering
University of Roma, Tor Vergata
00133 Roma, Italy
{croce,annesi,storch,basili}@info.uniroma2.it

Abstract

This paper presents the UNITOR system that participated to the SemEval 2012 Task 6: Semantic Textual Similarity (STS). The task is here modeled as a Support Vector (SV) regression problem, where a similarity scoring function between text pairs is acquired from examples. The semantic relatedness between sentences is modeled in an unsupervised fashion through different similarity functions, each capturing a specific semantic aspect of the STS, e.g. syntactic vs. lexical or topical vs. paradigmatic similarity. The SV regressor effectively combines the different models, learning a scoring function that weights individual scores in a unique resulting STS. It provides a highly portable method as it does not depend on any manually built resource (e.g. WordNet) nor controlled, e.g. aligned, corpus.

1 Introduction

Semantic Textual Similarity (STS) measures the degree of semantic equivalence between two phrases or texts. An effective method to compute similarity between short texts or sentences has many applications in Natural Language Processing (Mihalcea et al., 2006) and related areas such as Information Retrieval, e.g. to improve the effectiveness of a semantic search engine (Sahami and Heilman, 2006), or databases, where text similarity can be used in schema matching to solve semantic heterogeneity (Islam and Inkpen, 2008).

STS is here modeled as a Support Vector (SV) regression problem, where a SV regressor learns the similarity function over text pairs. Regression learning has been already applied to different NLP tasks. In (Pang and Lee, 2005) it is applied to Opinion Mining, in particular to the rating-inference problem, wherein one must determine an author evaluation with respect to a multi-point scale. In (Albrecht and Hwa, 2007) a method is proposed for developing sentence-level MT evaluation metrics using regression learning without directly relying on human reference translations. In (Biadsy et al., 2008) it has been used to rank candidate sentences for the task of producing biographies from Wikipedia. Finally, in (Becker et al., 2011) SV regressor has been used to rank questions within their context in the multimodal tutorial dialogue problem.

In this paper, the semantic relatedness between two sentences is modeled as a combination of different similarity functions, each describing the analogy between the two texts according to a specific semantic perspective: in this way, we aim at capturing syntactic and lexical equivalences between sentences and exploiting either topical relatedness or paradigmatic similarity between individual words. The variety of semantic evidences that a system can employ here grows quickly, according to the genre and complexity of the targeted sentences. We thus propose to combine such a body of evidence to learn a comprehensive scoring function $y = f(\vec{x})$ over individual measures from labeled data through SV regression: y is the gold similarity score (provided by human annotators), while $\vec{x}$ is the vector of the different individual scores, provided by the chosen similarity functions. The regressor objective is to learn the proper combination of different functions redundantly applied in an unsupervised fashion, without involving any in-depth description of the target domain or prior knowledge. The resulting function selects and filters the most useful information and it

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 597–602,
Montréal, Canada, June 7-8, 2012.

is a highly portable method. In fact, it does not depend on manually built resources (e.g. WordNet), but mainly exploits distributional analysis of unlabeled corpora.

In Section 2, the employed similarity functions are described and the application of SV regression is presented. Finally, Section 3 discusses results on the SemEval 2012 - Task 6.

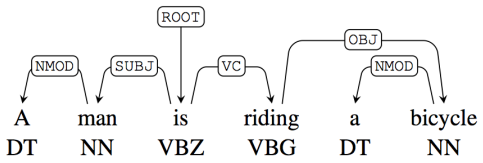

Figure 1: Example of dependency graph

2 Combining different similarity function through SV regression

This section describes the UNITOR systems participating to the SemEval 2012 Task 6: in Section 2.1 the different similarity functions between sentence pairs are discussed, while Section 2.2 describes how the SV regression learning is applied.

2.1 STS functions

Each STS depends on a variety of linguistic aspects in data, e.g. syntactic or lexical information. While their supervised combination can be derived through SV regression, different unsupervised estimators of STS exist.

Lexical Overlap (LO). A basic similarity function is first employed as the *lexical overlap between sentences*, i.e. the cardinality of the set of words occurring in both sentences.

Document-oriented similarity based on Latent Semantic Analysis (LSA). This function captures latent semantic topics through LSA. The adjacency terms-by-documents matrix is first acquired through the distributional analysis of a corpus and reduced through the application of Singular Value Decomposition (SVD), as described in (Landauer and Dumais, 1997). In this work, the individual sentences are assumed as pseudo documents and represented by vectors in the lower dimensional LSA space. The cosine similarity between vectors of a sentence pair is the metric hereafter referred to as *topical similarity*.

Compositional Distributional Semantics (CDS). Lexical similarity can also be extended to account for syntactic compositions between words. This makes sentence similarity to depend on the set of individual compounds, e.g. subject-verb relationship instances. While basic lexical information can still be obtained by distributional analysis, phrase level similarity can be here modeled as a specific function of the co-occurring words, i.e. a complex algebraic composition of their corresponding word vectors. Differently from the document-oriented case used in the LSA function, base lexical vectors are here derived from co-occurrence counts in a word space, built according to the method discussed in (Sahlgren, 2006; Croce and Previtali, 2010). In order to keep dimensionality as low as possible, SVD is also applied here (Annesi et al., 2012). The result is that every noun, verb, adjective and adverb is then projected in the reduced word space and then different composition functions can be applied as discussed in (Mitchell and Lapata, 2010) or (Annesi et al., 2012).

Convolution kernel-based similarity. The similarity function is here the Smoothed Partial Tree Kernel (**SPTK**) proposed in (Croce et al., 2011). This convolution kernel estimates the similarity between sentences, according to the syntactic and lexical information in both sentences. Syntactic representation of a sentence like "*A man is riding a bicycle*" is derived from the dependency parse tree, as shown in Fig. 1. It allows to define different tree structures over which the SPTK operates. First, a tree including only lexemes, where edges encode their dependencies, is generated and called Lexical Only Centered Tree (LOCT), see Fig. 2. Then, we add to each lexical node two leftmost children, encoding the grammatical function and the POS-Tag respectively: it is the so-called Lexical Centered Tree (LCT), see Fig. 3. Finally, we generate the Grammatical Relation Centered Tree (GRCT), see Fig. 4, by setting grammatical relation as non-terminal nodes, while PoS-Tags are pre-terminals and fathers of their associated lexemes. Each tree representation provides a different kernel function so that three different SPTK similarity scores, i.e. LOCT, LCT and GRCT, are here obtained.

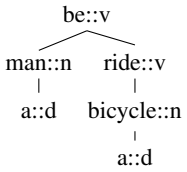

Figure 2: Lexical Only Centered Tree (LOCT)

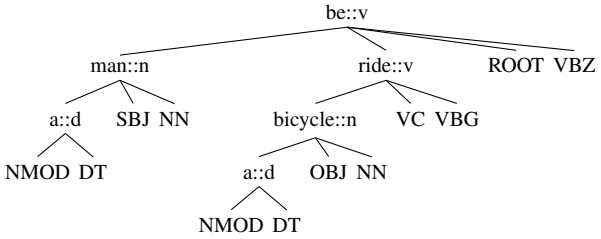

Figure 3: Lexical Centered Tree (LCT)

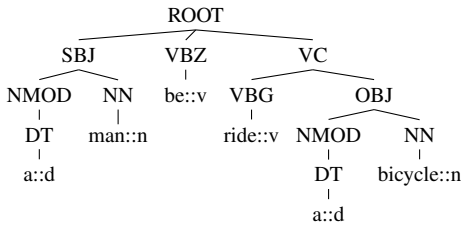

Figure 4: Grammatical Relation Centered Tree (GRCT)

2.2 Combining STSs with SV Regression

The similarity functions described above provide scores capturing different linguistic aspects and an effective way to combine such information is made available by Support Vector (SV) regression, described in (Smola and Schölkopf, 2004). The idea is to learn a higher level model by weighting scores according to specific needs implicit in training data. Given similarity scores $\vec{x}_i$ for the i-th sentence pair, the regressor learns a function $y_i = f(\vec{x}_i)$, where y_i is the score provided by human annotators.

The ε-SV regression (Vapnik, 1995) algorithm allows to define the best f approximating the training data, i.e. the function that has at most ε deviation from the actually obtained targets y_i for all the training data. Given a training dataset $\{(\vec{x}_1, y_1), \ldots, (\vec{x}_l, y_l)\} \in X \times \mathbb{R}$, where X is the space of the input patterns, i.e. the original similarity scores, we can acquire a linear function

$$f(\vec{x}) = \langle \vec{w}, \vec{x} \rangle + b \text{ with } \vec{w} \in X, b \in \mathbb{R}$$

by solving the following optimization problem:

$$\text{minimize } \frac{1}{2}||\vec{w}||^2$$
$$\text{subject to } \begin{cases} y_i - \langle \vec{w}, \vec{x}_i \rangle - b \leq \varepsilon \\ \langle \vec{w}, \vec{x}_i \rangle + b - y_i \leq \varepsilon \end{cases}$$

Since the function f approximating all pairs $(\vec{x}_i, y_i)$ with ε precision, may not exist, i.e. the convex optimization problem is infeasible, slack variables ξ_i, ξ_i^* are introduced:

$$\text{minimize } \frac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$$
$$\text{subject to } \begin{cases} y_i - \langle \vec{w}, \vec{x}_i \rangle - b \leq \varepsilon + \xi_i \\ \langle \vec{w}, \vec{x}_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

where ξ_i, ξ_i^* measure the error introduced by training data with a deviation higher than ε and the constant $C > 0$ determines the trade-off between the norm $\|\vec{w}\|$ and the amount up to which deviations larger than ε are tolerated.

3 Experimental Evaluation

This section describes results obtained in the SemEval 2012 Task 6: STS. First, the experimental setup of different similarity functions is described. Then, results obtained over training datasets are reported. Finally, results achieved in the competition are discussed.

3.1 Experimental setup

In order to estimate the Latent Semantic Analysis (LSA) based similarity function, the distributional analysis of the English version of the Europarl Corpus (Koehn, 2002) has been carried out. It is the same source corpus of the SMTeuroparl dataset and it allows to acquire a semantic space capturing the same topics characterizing this dataset. A word-by-sentence matrix models the sentence representation space. The entire corpus has been split so that each vector represents a sentence: the number of different sentences is about 1.8 million and the matrix cells contain *tf-idf* scores between words and sentences. The SVD is applied and the space dimensionality

is reduced to $k = 250$. Novel sentences are immersed in the reduced space, as described in (Landauer and Dumais, 1997) and the LSA-based similarity between two sentences is estimated according the cosine similarity.

To estimate the Compositional Distributional Semantics (CDS) based function, a co-occurrence Word Space is first acquired through the distributional analysis of the UKWaC corpus (Baroni et al., 2009), i.e. a Web document collection made of about 2 billion tokens. UKWaC is larger than the Europarl corpus and we expect it makes available a more general lexical representation suited for all datasets. An approach similar to the one described in (Croce and Previtali, 2010) has been adopted for the acquisition of the word space. First, all words occurring more than 200 times (i.e. the *targets*) are represented through vectors. The original space dimensions are generated from the set of the 20,000 most frequent words (i.e. *features*) in the UKWaC corpus. One dimension describes the Pointwise Mutual Information score between one feature as it occurs on a left or right window of 3 tokens around a target. Left contexts of targets are treated differently from the right ones, in order to also capture asymmetric syntactic behaviors (e.g., useful for verbs): 40,000 dimensional vectors are thus derived for each target. The particularly small window size allows to better capture paradigmatic relations between targets, e.g. hyponymy or synonymy. Again, the SVD reduction is applied to the original matrix with a $k = 250$. Once lexical vectors are available, a compositional similarity measure can be obtained by combining the word vectors according to a CDS operator, e.g. (Mitchell and Lapata, 2010) or (Annesi et al., 2012). In this work, the adopted compositional representation is the *additive* operator between lexical vectors, as described in (Mitchell and Lapata, 2010) and the similarity function between two sentences is the cosine similarity between their corresponding compositional vectors. Moreover, two additive operators that only sum over nouns and verbs are also adopted, denoted by CDS_V and CDS_N, respectively.

The estimation of the semantically Smoothed Partial Tree Kernel (SPTK) is made available by an extended version of SVM-LightTK software[1] (Moschitti, 2006) implementing the smooth matching between tree nodes. The tree representation described in Sec. 2.1 allows to define 3 different kernels, i.e. SPTK_{LOCT}, SPTK_{LCT} and SPTK_{GRCT}. Similarity between lexical nodes is estimated as the cosine similarity in the co-occurrence Word Space described above, as in (Croce et al., 2011).

In all corpus analysis and experiments, sentences are processed with the LTH dependency parser, described in (Johansson and Nugues, 2007), for Part-of-speech tagging and lemmatization. Dependency parsing of datasets is required for the SPTK application. Finally, SVM-LightTK is employed for the SV regression learning to combine specific similarity functions.

3.2 Evaluating the impact of unsupervised models

Table 1 compares the Pearson Correlation of different similarity functions described in Section 2.1, i.e. mainly the results of the unsupervised approaches, against the challenge training data. Regarding to MSRvid dataset, the topical similarity (LSA function) achieves the best result, i.e. 0.748. Paradigmatic lexical information as in CDS, CDS_N and LO provides also good results, confirming the impact of lexical generalization. However, only nouns seem to contribute significantly, as for the poor results of CDS_V suggest. As the dataset is characterized by short sentences with negligible syntactic differences, SPTK-based kernels are not discriminant. On the contrary, the SPTK_{LCT} achieves the best result in the MSRpar dataset, where paraphrasing phenomena are peculiar. Notice that the other SPTK kernels are not equivalently performant, in line with previous results on question classification and semantic role labeling (Croce et al., 2011). Lexical information provides a crucial contribution also for LO, although the contribution of topical or paradigmatic generalization seems negligible over MSRpar. Finally, in the SMTeuroparl, longer sentences are the norm and length seems to compromise the performance of LO. The best results seem to require the lexical and syntactic information provided by CDS and SPTK.

[1] http://disi.unitn.it/moschitti/Tree-Kernel.htm

Models	Dataset		
	MSRvid	MSRpar	SMTeuroparl
CDS	.652	.393	**.681**
CDS_N	.630	.234	.485
CDS_V	.219	.317	.264
LSA	**.748**	.344	.477
$SPTK_{LOCT}$	.300	.251	.611
$SPTK_{LCT}$	.297	**.464**	.622
$SPTK_{GRCT}$	.278	.255	.626
LO	.560	.446	.248

Table 1: Unsupervised results over the training dataset

3.3 Evaluating the role of SV regression

The SV regressors have been trained over a feature space that enumerates the different similarity functions: one feature is provided by the LSA function, three by the CDS, i.e. CDS, CDS_N and CDS_V, three by SPTK, i.e. $SPTK_{LOCT}$, $SPTK_{LCT}$ and $SPTK_{GRCT}$ and one by LO, i.e. the number of words in common. Two more features are obtained by the *sentence lengths* of a pair, i.e. the number of words in the first and second sentence, respectively. Table 2 shows Pearson Correlation results when the regressor is trained according a 10-fold cross validation schema. First, all possible feature combinations are attempted for the SV regression, so that every subset of the 10 features is evaluated. Results of the best feature combination are shown in column $best_{feat}$: for MSRvid, the best performance is achieved when all 10 features are considered; in MSRpar, SPTK combined with LO is sufficient; finally, in the SMTeuroparl the combination is LO, CDS and SPTK. In column all_{feat} results achieved by considering all features are reported. Last column specifies the performance increase with respect to the corresponding best results in the unsupervised settings.

Results of the regressors are always higher with respect to the unsupervised settings, with up to a 35% improvement for the MSRpar, i.e. the most complex domain. Moreover, differences when best and all features are employed are negligible. It means that SV regressor allows to automatically combine and select the most informative similarity aspects, confirming the applicability of the proposed redundant approach to STS.

Dataset	Experiments		Gain
	$best_{feat}$	all_{feat}	
MSRvid	.789	.789	5,0%
MSRpar	.615	.612	32,4%
SMTeuroparl	.692	.691	1,6%

Table 2: SV regressor results over the training dataset

3.4 Results over the SemEval Task 6

According to the above evidence, we participated to the SemEval challenge with three different systems.

Sys_1 - Best Features. Scores between pairs from a specific dataset are obtained by applying a regressor trained over pairs from the same dataset. It means that, for example, the test pairs from the MSRvid dataset are processed with a regressor trained over the MSRvid training data. Moreover, the most representative similarity function estimated for the collection is employed: the feature combination providing the best correlation results over training pairs is adopted for the test. The same is applied to MSRpar and SMTeuroparl. No selection is adopted for the Surprise data and training data for all the domains are used, as described in Sys_3.

Sys_2 - All Features. Relatedness scores between pairs from a specific dataset are obtained using a regressor trained using pairs from the same dataset. Differently from the Sys_1, the similarity function here is employed within the SV regressors trained over all 10 similarity functions (i.e. all features).

Sys_3 - All features and All domains. The SV regressor is trained using training pairs from *all* collections and over *all* 10 features. It means that one single model is trained and employed to score all test data. This approach is also used for the Surprise data, i.e. the OnWN and SMTnews datasets.

Table 3 reports the general outcome for the UNITOR systems. Rank of the individual scores with respect to the other systems participating to the challenge is reported in parenthesis. This allows to draw some conclusions. First, the proposed system ranks around the 12 and 13 system positions (out of 89 systems), and the 6th group. The adoption of all proposed features suggests that more evidence is better, as it can be properly modeled by regression. It seems generally better suited for the variety of semantic phenomena observed in the tests. Regressors seem

Dataset	Results			
	BL	Sys_1	Sys_2	Sys_3
MSRvid	.299	**.821**	**.821**	.802
MSRpar	.433	.569	**.576**	.468
SMTeuroparl	.454	**.516**	.510	.457
surp.OnWN	.586		**.659**	
surp.SMTnews	.390		**.471**	
ALL	.311	.747 (13)	**.747 (12)**	.628 (40)
ALLnrm	.673	.829 (12)	**.830 (11)**	.815 (21)
Mean	.436	.632 (10)	**.632 (9)**	.594 (28)

Table 3: Results over the challenge test dataset

to be robust enough to select the proper features and make the feature selection step (through collection specific cross-validation) useless. Collection specific training seems useful, as Sys_3 achieves lower results, basically due to the significant stylistic differences across the collections. However, the good level of accuracy achieved over the surprise data sets (between 11% and 17% performance gain with respect to the baselines) confirms the large applicability of the overall technique: our system in fact does not depend on *any* manually coded resource (e.g. WordNet) nor on any controlled (e.g. parallel or aligned) corpus. Future work includes the study of the learning rate and its correlation with different and richer similarity functions, e.g. CDS as in (Annesi et al., 2012).

Acknowledgements This research is partially supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant numbers 262491 (INSEARCH). Many thanks to the reviewers for their valuable suggestions.

References

Joshua Albrecht and Rebecca Hwa. 2007. Regression for sentence-level mt evaluation with pseudo references. In *Proceedings of ACL*, pages 296–303, Prague, Czech Republic, June.

Paolo Annesi, Valerio Storch, and Roberto Basili. 2012. Space projections as distributional models for semantic composition. In *CICLing (1)*, Lecture Notes in Computer Science, pages 323–335. Springer.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Lee Becker, Martha Palmer, Sarel van Vuuren, and Wayne Ward. 2011. Evaluating questions in context.

Fadi Biadsy, Julia Hirschberg, and Elena Filatova. 2008. An unsupervised approach to biography production using wikipedia. In *ACL*, pages 807–815.

Danilo Croce and Daniele Previtali. 2010. Manifold learning for the semi-supervised induction of framenet predicates: An empirical investigation. In *Proceedings of the GEMS 2010 Workshop*, pages 7–16, Uppsala, Sweden.

Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2011. Structured lexical similarity via convolution kernels on dependency trees. In *Proceedings of EMNLP*, Edinburgh, Scotland, UK.

Aminul Islam and Diana Inkpen. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Trans. Knowl. Discov. Data*, 2:10:1–10:25, July.

Richard Johansson and Pierre Nugues. 2007. Semantic structure extraction using nonprojective dependency trees. In *Proceedings of SemEval-2007*, Prague, Czech Republic, June 23-24.

P. Koehn. 2002. Europarl: A multilingual corpus for evaluation of machine translation. *Draft*.

Thomas K Landauer and Susan T. Dumais. 1997. A solution to platos problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, pages 211–240.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *In AAAI06*.

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

Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of ECML'06*, pages 318–329.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*.

Mehran Sahami and Timothy D. Heilman. 2006. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 377–386, New York, NY, USA. ACM.

Magnus Sahlgren. 2006. *The Word-Space Model*. Ph.D. thesis, Stockholm University.

Alex J. Smola and Bernhard Schölkopf. 2004. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August.

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer–Verlag, New York.