UNITOR-CORE_TYPED: Combining Text Similarity and Semantic Filters through SV Regression

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

Abstract

This paper presents the UNITOR system that participated in the *SEM 2013 shared task on Semantic Textual Similarity (STS). The task is modeled as a Support Vector (SV) regression problem, where a similarity scoring function between text pairs is acquired from examples. The proposed approach has been implemented in a system that aims at providing high applicability and robustness, in order to reduce the risk of over-fitting over a specific datasets. Moreover, the approach does not require any manually coded resource (e.g. WordNet), but mainly exploits distributional analysis of unlabeled corpora. A good level of accuracy is achieved over the shared task: in the Typed STS task the proposed system ranks in 1st and 2nd position.

1 Introduction

Semantic Textual Similarity (STS) measures the degree of semantic equivalence between two phrases or texts. An effective method to compute similarity between sentences or semi-structured material has many applications in Natural Language Processing (Mihalcea et al., 2006) and related areas such as Information Retrieval, improving the effectiveness of semantic search engines (Sahami and Heilman, 2006), or databases, using text similarity in schema matching to solve semantic heterogeneity (Islam and Inkpen, 2008).

This paper describes the UNITOR system participating in both tasks of the *SEM 2013 shared task on Semantic Textual Similarity (STS), described in (Agirre et al., 2013):

- the **Core STS tasks**: given two sentences, s_1 and s_2, participants are asked to provide a score reflecting the corresponding text similarity. It is the same task proposed in (Agirre et al., 2012).
- the **Typed-similarity STS task**: given two semi-structured records t_1 and t_2, containing several typed fields with textual values, participants are asked to provide multiple similarity scores: the types of similarity to be studied include *location*, *author*, *people involved*, *time*, *events or actions*, *subject* and *description*.

In line with several participants of the STS 2012 challenge, such as (Banea et al., 2012; Croce et al., 2012a; Šarić et al., 2012), STS is here modeled as a Support Vector (SV) regression problem, where a SV regressor learns the similarity function over text pairs. The semantic relatedness between two sentences is first modeled in an unsupervised fashion by several similarity functions, each describing the analogy between the two texts according to a specific semantic perspective. We aim at capturing separately syntactic and lexical equivalences between sentences and exploiting either topical relatedness or paradigmatic similarity between individual words. Such information is then combined in a supervised schema through a 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.

For the Typed STS task, given the specificity of the involved information and the heterogeneity of target scores, individual measures are not applied to entire texts. Specific phrases are filtered according to linguistic policies, e.g. words characterized by specific Part-of-Speech (POS), such as nouns and verbs, or Named Entity (NE) Category, i.e. men-

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task*, pages 59–65, Atlanta, Georgia, June 13-14, 2013. ©2013 Association for Computational Linguistics

tions to specific name classes, such as of a PERSON, LOCATION or DATE. The former allows to focus the similarity functions over entities (nouns) or actions (verbs), while the latter allows to focus on some aspects connected with the targeted similarity functions, such as *person_involved*, *location* or *time*.

The proposed approach has been implemented in a system that aims at providing high applicability and robustness. This objective is pursued by adopting four similarity measures designed to avoid the risk of over-fitting over each specific dataset. Moreover, the approach does not require any manually coded resource (e.g. WordNet), but mainly exploits distributional analysis of unlabeled corpora. Despite of its simplicity, a good level of accuracy is achieved over the 2013 STS challenge: in the Typed STS task the proposed system ranks 1^{st} and 2^{nd} position (out of 18); in the Core STS task, it ranks around the 37^{th} position (out of 90) and a simple refinement to our model makes it 19^{th}.

In the rest of the paper, in Section 2, the employed similarity functions are described and the application of SV regression is presented. Finally, Section 3 discusses results on the *SEM 2013 shared task.

2 Similarity functions, regression and linguistic filtering

This section describes the approach behind the UNITOR system. The basic similarity functions and their combination via SV regressor are discussed in Section 2.1, while the linguistic filters are presented in Section 2.2.

2.1 STS functions

Each STS function 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. A basic similarity function is modeled as the *Lexical Overlap* (**LO**) between sentences. Given the sets W_a and W_b of words occurring in two generic texts t_a and t_b, LO is estimated as the *Jaccard Similarity* between the sets, i.e. LO=$\frac{|W_a \cap W_b|}{|W_a \cup W_b|}$. In order to reduce data sparseness, lemmatization is applied and each word is enriched with its POS to avoid the confusion between words from different grammatical classes.

Compositional Distributional Semantics. Other similarity functions are obtained by accounting for the syntactic composition of the lexical information involved in the sentences. Basic lexical information is obtained by a co-occurrence Word Space that is built according to (Sahlgren, 2006; Croce and Previtali, 2010). Every word appearing in a sentence is then projected in such space. A sentence can be thus represented neglecting its syntactic structure, by applying an *additive linear combination*, i.e. the so-called **SUM** operator. The similarity function between two sentences is then the cosine similarity between their corresponding vectors.

A second function is obtained by applying a *Distributional Compositional Semantics* operator, in line with the approaches introduced in (Mitchell and Lapata, 2010), and it is adopted to account for semantic composition. In particular, the approach described in (Croce et al., 2012c) has been applied. It is based on space projection operations over basic geometric lexical representations: syntactic bigrams are projected in the so called *Support Subspace* (Annesi et al., 2012), aimed at emphasizing the semantic features shared by the compound words. The aim is to model semantics of syntactic bi-grams as projections in lexically-driven subspaces. In order to extend this approach to handle entire sentences, we need to convert them in syntactic representations compatible with the compositional operators proposed. A dependency grammar based formalism captures binary syntactic relations between the words, expressed as nodes in a dependency graph. Given a sentence, the parse structure is acquired and different triples (w_1, w_2, r) are generated, where w_1 is the relation governor, w_2 is the dependent and r is the grammatical type. In (Croce et al., 2012c) a simple approach is defined, and it is inspired by the notion of Soft Cardinality, (Jimenez et al., 2012). Given a triple set $T = \{t_1, \ldots, t_n\}$ extracted from a sentence S and a similarity $sim(t_i, t_j)$, the Soft Cardinality is estimated as $|S|'_{sim} \approxeq \sum_{t_i}^{|T|}(\sum_{t_j}^{|T|} sim(t_i, t_j)^p)^{-1}$, where parameter p controls the “softness” of the cardinality: with $p = 1$ element similarities are unchanged while higher value will tend to the Classical Cardinality measure. Notice that differently from the previous usage of the Soft Cardinality notion, we did not apply it to sets of individual words, but to the sets of dependencies (i.e. triples) derived from the two sentences. The sim function here can be thus replaced by any compositional operator among the ones discussed in (Annesi et al., 2012). Given two sentences, higher Soft Cardinality values mean that the elements in both sentences (i.e. triples) are different, while the lower values mean that common triples are identical or very similar, suggesting that sentences contain the same kind of information. Given the sets of triples A and B extracted from the two candidate sentences, our approach estimates a syntactically restricted soft cardinality operator, the *Syntactic Soft Cardinality* (**SSC**) as $SSC(A,B) = \frac{2|A\cap B|'}{|A|'+|B|'}$, as a "soft approximation" of Dice's coefficient calculated on both sets[1].

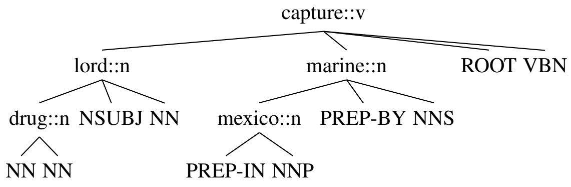

Figure 1: Lexical Centered Tree (LCT)

Convolution kernel-based similarity. The similarity function is here the *Smoothed Partial Tree Kernel* (**SPTK**) proposed in (Croce et al., 2011). SPTK is a generalized formulation of a Convolution Kernel function (Haussler, 1999), i.e. the Tree Kernel (TK), by extending the similarity between tree structures with a function of node similarity. The main characteristic of SPTK is its ability to measure the similarity between syntactic tree structures, which are partially similar and whose nodes can differ but are semantically related. One of the most important outcomes is that SPTK allows "embedding" external lexical information in the kernel function only through a similarity function among lexical nodes, namely words. Moreover, SPTK only requires this similarity to be a valid kernel itself. This means that such lexical information can be derived from lexical resources or it can be automatically acquired by a Word Space. The SPTK is applied to a specific tree representation that allowed to achieve state-of-the-art results on several complex semantic tasks, such as Question Classification (Croce et al., 2011) or Verb Classification (Croce et al., 2012b): each sentence is represented through the Lexical Centered Tree (LCT), as shown in Figure 1 for the sentence "*Drug lord captured by Marines in Mexico*". It is derived from the dependency parse tree: nodes reflect lexemes and edges encode their syntactic dependencies; then, we add to each lexical node two leftmost children, encoding the grammatical function and the POS-Tag respectively.

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. Moreover, since the combination of kernel is still a kernel, we can apply polynomial and RBF kernels (Shawe-Taylor and Cristianini, 2004) to the regressor.

2.2 Semantic constraints for the Typed STS

Typed STS insists on records, i.e. sequence of typed textual fields, rather than on individual sentences. Our aim is to model the typed task with the same spirit as the core one, through a combination of different linguistic evidences, which are modeled through independent kernels. The overall similarity model described in 2.1 has been thus applied also to the typed task according to two main model changes:

- **Semantic Modeling**. Although SV regression is still applied to model one similarity type, each type depends on a subset of the multiple evidences originating from individual fields: one similarity type acts as a filter on the set of fields, on which kernels will be then applied.

- **Learning Constraints**. The selected *fields* provide different evidences to the regression steps. Correspondingly, each similarity type corresponds to specific kernels and features for its fields. These constraints are applied by selecting features and kernels for each field.

[1] Notice that, since the intersection $|A \cap B|'$ tends to be too strict, we approximate it from the union cardinality estimation $|A|' + |B|' - |A \cup B|'$.

	`dcTitle`	`dcSubject`	`dcDescription`	`dcCreator`	`dcDate`	`dcSource`
author	-	-	Per	$*$	-	-
people_inv.	Per	Per	Per	-	-	-
time	Date	Date	Date	-	$*$	-
location	Loc	Loc	Loc	-	-	-
event	$N, V, N \cup V$	$N, V, N \cup V$	$N, V, N \cup V$	-	-	-
subject	$N, V, J, N \cup J \cup V$	$N, V, J, N \cup J \cup V$	-	-	-	-
description	-	-	$N, V, J, N \cup J \cup V$	-	-	-
general	$+$	$+$	$+$	$*$	$*$	$*$

Table 1: Filtering Schema adopted for the Typed STS task.

Notice how some kernels loose significance in the typed STS task. Syntactic information is no useful so that no tree kernel and compositional kernel is applied here. Most of the fields are non-sentential[2]. Moreover, not all morpho-syntactic information are extracted as feature from some fields. Filters usually specify some syntactic categories or Named Entities (NEs): they are textual mentions to specific real-world categories, such as of Persons (Per), Locations (Loc) or Dates. They are detected in a field and made available as feature to the corresponding kernel: this introduces a bias on typed measures and emphasizes specific semantic aspects (e.g. places Loc or persons Per, in *location* or *author* measures, respectively). For example, in the sentence "*The chemist R.S. Hudson began manufacturing soap in the back of his small shop in West Bomich in 1837*", when POS tag filters are applied, only verbs (*V*), nouns (*N*) or adjectives (*J*) can be selected as features. This allows to focus on specific actions, e.g. the verb "*manufacture*", entities, e.g. nouns "*soap*" and "*shop*", or some properties, e.g. the adjective "*small*". When Named Entity categories are used, a mention to a person like "*R.S. Hudson*" or to a location, e.g. "*West Bomich*', or date, e.g. "*1837*", can be useful to model the the *person_involved*, the *location* or *time* similarity measures, respectively.

The Semantic Modeling and the Learning Constraints system adopted to model the Typed STS task are defined in Table 1. There rows are the different target similarities, while columns indicate document fields, such as `dcTitle`, `dcSubject`, `dcDescription`, `dcCreator`, `dcDate` and `dcSource`, as described in the *SEM 2013 shared task description. Each entry in the Table represents the feature set for that fields, i.e. POS tags (i.e. V, N, J) or Named Entity classes. The "$*$" symbol corresponds to all features, i.e. no restriction is applied to any POS tag or NE class. Finally, the *general* similarity function makes use of every NE class and POS tags adopted for that field in any measure, as expressed by the special notation $+$, i.e. "*all of the above features*".

Every feature set denoted in the Table 1 supports the application of a lexical kernel, such as the LO described in Section 2.1. When different POS tags are requested (such as N and V) multiple feature sets and kernels are made available. The "-" symbol means that the source field is fully neglected from the SV regression. As an example, the SV regressor for the *location* similarity has been acquired considering the fields `dcTitle`, `dcSubject`, `dcDescription`. Only features used for the kernel correspond to Locations (Loc). For each of the three feature, the LO and SUM similarity function has been applied, giving rise to an input 6-dimensional feature space for the regressor. Differently, in the *subject* similarity, nouns, adjectives and verbs are the only features adopted from the fields `dcSubject`, `dcTitle`, so that 8 feature sets are used to model these fields, giving rise to a 16-dimensional feature space.

3 Results and discussion

This section describes results obtained in the *SEM 2013 shared task. The experimental setup of different similarity functions is described in Section 3.1. Results obtained over the Core STS task and Typed STS task are described in Section 3.2 and 3.3.

[2] The `dcDescription` is also made of multiple sentences and it reduces the applicability of SPTK and SSC: parse trees have no clear alignment.

3.1 Experimental setup

In all experiments, sentences are processed with the Stanford CoreNLP[3] system, for Part-of-Speech tagging, lemmatization, named entity recognition[4] and dependency parsing.

In order to estimate the basic lexical similarity function employed in the SUM, SSC and SPTK operators, a co-occurrence Word Space is acquired through the distributional analysis of the UkWaC corpus (Baroni et al., 2009), a Web document collection made of about 2 billion tokens. The same setting of (Croce et al., 2012a) has been adopted for the space acquisition. The same setup described in (Croce et al., 2012c) is applied to estimate the SSC function. The similarity between pairs of syntactically restricted word compound is evaluated through a *Symmetric model*: it selects the best 200 dimensions of the space, selected by maximizing the component-wise product of each compound as in (Annesi et al., 2012), and combines the similarity scores measured in each couple subspace with the product function. The similarity score in each subspace is obtained by summing the cosine similarity of the corresponding projected words. The "soft cardinality" is estimated with the parameter $p = 2$.

The estimation of the semantically Smoothed Partial Tree Kernel (SPTK) is made available by an extended version of SVM-LightTK software[5] (Moschitti, 2006) implementing the smooth matching between tree nodes. Similarity between lexical nodes is estimated as the cosine similarity in the co-occurrence Word Space described above, as in (Croce et al., 2011). Finally, SVM-LightTK is employed for the SV regression learning to combine specific similarity functions.

3.2 Results over the Core STS

In the Core STS task, the resulting text similarity score is measured by the regressor: each sentence pair from all datasets is modeled according to a 13 dimensional feature space derived from the different functions introduced in Section 2.1, as follows.

The first 5 dimensions are derived by applying the LO operator over lemmatized words in the noun, verb, adjective and adverb POS categories: 4 kernels look at individual categories, while a fifth kernel insists on the union of all POS. A second set of 5 dimensions is derived by the same application of the SUM operator to the same syntactic selection of features. The SPTK is then applied to estimate the similarity between the LCT structures derived from the dependency parse trees of sentences. Then, the SPTK is applied to derive an additional score without considering any specific similarity function between lexical nodes; in this setting, the SPTK can be considered as a traditional Partial Tree Kernel (Moschitti, 2006), in order to capture a more strict syntactical similarity between texts. The last score is generated by applying the SSC operator.

	Run$_1$	Run$_2$	Run$_3$	Run$_1^*$
headlines	.635 (50)	**.651 (39)**	.603 (58)	.671 (30)
OnWN	**.574 (33)**	.561 (36)	.549 (40)	.637 (25)
FNWN	.352 (35)	**.358 (32)**	.327 (44)	.459 (7)
SMT	**.328 (39)**	.310 (49)	.319 (44)	.348 (21)
Mean	**.494 (37)**	.490 (42)	.472 (52)	.537 (19)

Table 2: Results over the Core STS task

We participated in the *SEM challenge with three different runs. The main difference between each run is the dataset employed in the training phase and the employed kernel within the regressor. Without any specific information about the test datasets, a strategy to prevent the regressor to over-fit training material has been applied. We decided to use a training dataset that achieved the best results over datasets radically different from the training material in the STS challenge of Semeval 2012. In particular, for the FNWN and OnWN datasets, we arbitrarily selected the training material achieving best results over the 2012 surprise.OnWN; for the headlines and SMT datasets we maximized performance training over surprise.SMTnews. In **Run$_1$** the SVM regressor is trained using dataset combinations providing best results according to the above criteria: MSRpar, MSRvid, SMTeuroparl and surprise.OnWN are employed against FNWN and OnWN; MSRpar, SMTeuroparl and surprise.SMTnews are employed against headline and SMT. A linear kernel is applied when training the regressor. In **Run$_2$**, differently from the previous one, the SVM regressor is

[3] http://nlp.stanford.edu/software/corenlp.shtml

[4] The TIME and DURATION classes are collapsed with DATE, while the PERSON and LOCATION classes are considered without any modification.

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

	rank	general	author	people_inv.	time	location	event	subject	description	mean
Run_1	1	**.7981**	**.8158**	**.6922**	**.7471**	**.7723**	**.6835**	**.7875**	**.7996**	**.7620**
Run_2	2	.7564	.8076	.6758	.7090	.7351	.6623	.7520	.7745	.7341

Table 3: Results over the Typed STS task

trained using all examples from the training datasets. A linear kernel is applied when training the regressor. Finally, in $\mathbf{Run_3}$ the same training dataset selection schema of Run_1 is applied and a gaussian kernel is employed in the regressor.

Table 2 reports the general outcome for the UNITOR systems in term of Pearson Correlation. The best system, based on the linear kernel, ranks around the 35^{th} position (out of 90 systems), that reflects the mean rank of all the systems in the ranking of the different datasets. The gaussian kernel, employed for the Run_3 does not provide any contribution, as it ranks 50^{th}. We think that the main reason of these results is due to the intrinsic differences between training and testing datasets that have been heuristically coupled. This is first motivated by lower rank achieved by Run_2. Moreover, it is in line with the experimental findings of (Croce et al., 2012a), where a performance drop is shown when the regressor is trained over data that is not constrained over the corresponding source. In $\mathbf{Run_1^*}$ we thus optimized the system by manually selecting the training material that does provides best performance on the test dataset: MSRvid, SMTeuroparl and surprise.OnWN are employed against OnWN; surprise.OnWN against FNWN, SMTeuroparl against headlines; SMTeuroparl and surprise.SMTnews against SMT. A linear kernel within the regressor allow to reach the 19^{th} position, even reducing the complexity of the representation to a five dimensional feature space: LO and SUM without any specific filter, SPTK, PTK and SSC.

3.3 Results over the Typed STS

SV regression has been also applied to the Typed STS task through seven type-specific regressors plus a *general* one. Each SV regressor insists on the LO and SUM kernel as applied to the features in Table 1. Notice that it was mainly due to the lack of rich syntactic structures in almost all fields.

As described in Section 2.2, a specific modeling strategy has been applied to derive the feature space of each target similarity. For example, the regressor associated with the *event* similarity score is fed with 18 scores. Each of the 3 fields, , i.e. `dcTitle`, `dcSubject` and `dcDescription`, provides the 2 kernels (LO and SUM) with 3 feature sets (i.e. N, V and $N \cup V$). In particular, the *general* similarity function considers all extracted features for each field, giving rise to a space of 51 dimensions. We participated in the task with two different runs, whose main difference is the adopted kernel within the SV regressor. In $\mathbf{Run_1}$, a *linear kernel* is used, while in $\mathbf{Run_2}$ a *RBF kernel* is applied.

Table 3 reports the general outcome for the UNITOR system. The adopted semantic modeling, as well as the selection of the proper information, e.g. the proper named entity, allows the system to rank in the 1^{st} and 2^{nd} positions (out of 18 systems). The proposed selection schema in Table 1 is very effective, as confirmed by the results for almost all typed similarity scores. Again, the RBF kernel does not improve result over the linear kernel. The impact of the proposed approach can be noticed for very specific scores, such as *time* and *location*, especially for text pairs where structured information is absent, such as in the `dcDate` field. Moreover, the regressor is not affected by the differences between training and test dataset as for the previous Core STS task. A deep result analysis showed that some similarity scores are not correctly estimated within pairs showing partial similarities. For example, the *events or actions* typed similarity is overestimated for the texts pairs "*The Octagon and Pavilions, Pavilion Garden, Buxton, c 1875*" and "*The Beatles, The Octagon, Pavillion Gardens, St John's Road, Buxton, 1963*" because they mention the same location (i.e. "*Pavillion Gardens*").

Acknowledgements This work has been partially supported by the Regione Lazio under the project PROGRESS-IT (FILAS-CR-2011-1089) and the Italian Ministry of Industry within the "Industria 2015" Framework, project DIVINO (MI01_00234).

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012*, pages 385–393, Montréal, Canada, 7-8 June.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In **SEM 2013: The Second Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics.

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.

Carmen Banea, Samer Hassan, Michael Mohler, and Rada Mihalcea. 2012. Unt: A supervised synergistic approach to semantic text similarity. In **SEM 2012*, pages 635–642, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

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.

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.

Danilo Croce, Paolo Annesi, Valerio Storch, and Roberto Basili. 2012a. Unitor: Combining semantic text similarity functions through sv regression. In **SEM 2012*, pages 597–602, Montréal, Canada, 7-8 June.

Danilo Croce, Alessandro Moschitti, Roberto Basili, and Martha Palmer. 2012b. Verb classification using distributional similarity in syntactic and semantic structures. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 263–272, Jeju Island, Korea, July.

Danilo Croce, Valerio Storch, Paolo Annesi, and Roberto Basili. 2012c. Distributional compositional semantics and text similarity. *2012 IEEE Sixth International Conference on Semantic Computing*, 0:242–249.

David Haussler. 1999. Convolution kernels on discrete structures. Technical report, University of Santa Cruz.

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.

Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. 2012. Soft cardinality: A parameterized similarity function for text comparison. In **SEM 2012*, pages 449–453, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

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 *ECML*, pages 318–329, Berlin, Germany, September. Machine Learning: ECML 2006, 17th European Conference on Machine Learning, Proceedings.

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.

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA.

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

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. Takelab: Systems for measuring semantic text similarity. In **SEM 2012*, pages 441–448, Montréal, Canada, 7-8 June.