SERGIOJIMENEZ at SemEval-2016 Task 1: Effectively Combining Paraphrase Database, String Matching, WordNet and Word Embedding for Semantic Textual Similarity

Sergio Jimenez
Bogotá D.C., Colombia
sergio.jimenez.vargas@gmail.com

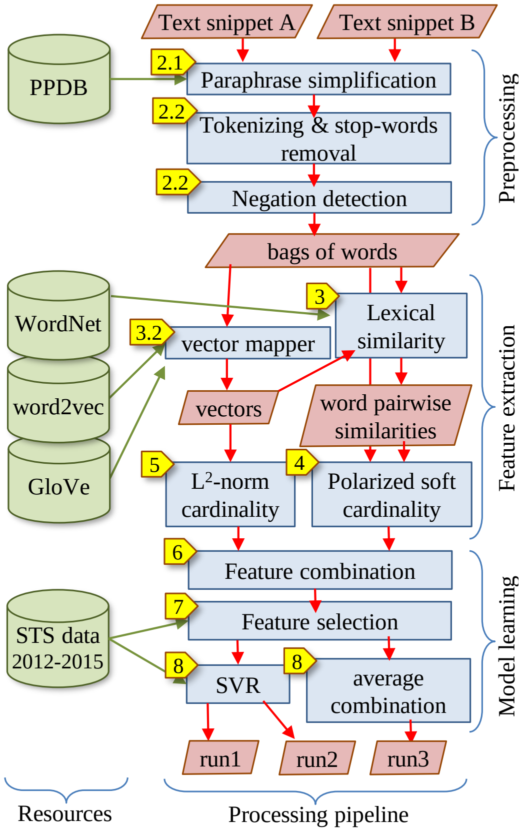
Abstract

In this paper, a system for semantic textual similarity, which participated in Task-1 in SemEval 2016 (monolingual and cross-lingual sub-tasks) is described. The system contains a preprocessing step that simplifies text using PPDB 2.0 and detects negations. Also, six lexical similarity functions were constructed using string matching, word embedding and synonyms-antonyms relations in WordNet. These lexical similarity functions are projected to sentence level using a new method called Polarized Soft Cardinality that supports negative similarities between words to model opposites. We also introduce a novel L^2-norm "cardinality" for vector space representations. The system extracts a set of 660 features from each pair of text snippets using the proposed cardinality measures. From this set, a subset of 12 features was selected in a supervised manner. These features are combined by SVR and, alternatively, by using the arithmetic mean to produce similarity predictions. Our team ranked second in the cross-lingual sub-task and got close to the best official results in the monolingual sub-task.

1 Introduction

Semantic Textual Similarity (STS) is a fundamental task in the field of natural language processing that has been addressed in SemEval competitions uninterruptedly since 2012 (Agirre et al., 2012; Agirre et al., 2013; Agirre et al., 2014; Agirre et al., 2015). The task is to compare two text fragments and produce a similarity score that is assessed according to human judgment. This year (Agirre et al., 2016), a new cross-lingual sub-task in English and Spanish is proposed in addition to the traditional monolingual English task. In SemEval 2015, the most popular approach among the best systems was the use of words alignments between sentences combining resources such as WordNet (Miller, George A., 1995), neural word embedding (Mikolov, Tomas et al., 2013) and the Paraphrase Database (Pavlick et al., 2015).

Figure 1: STS system architecture



Proceedings of SemEval-2016, pages 749–757,
San Diego, California, June 16-17, 2016. ©2016 Association for Computational Linguistics

This paper describes our system submission to STS 2016 that uses a cardinality-based approach (instead of word alignments) for combining the resources mentioned above. Several teams have used soft cardinality successfully in previous STS competitions from 2012 to 2014 (Jimenez et al., 2012; Jimenez et al., 2013a; Jimenez et al., 2013b; Jimenez et al., 2014; Lynum et al., 2014). For the proposed system, we extended the model of soft cardinality to allow the use of negative values in the lexical similarity component to model opposites due to antonymy and negation.

Figure 1 shows the overall architecture of the proposed system. Yellow labels in the upper left corner of each process component (blue squares) indicate the sections of this document where the module is discussed. In this figure, the processing pipeline is represented vertically in three stages: preprocessing, feature extraction and model learning. Red parallelograms represent the inputs and outputs of each process, from the pair of snippets of text for evaluation, through different intermediate representations (bag-of-words, vectors, etc.) and end on the predictions of similarity scores. The left side contains the used external resources linked to the process that makes use of each one.

2 Preprocessing

2.1 Paraphrase Simplification

The Paraphrase Database (PPDB) is a list of pairs of words, short phrases of syntactic rules where each pair is semantically equivalent in some degree (Pavlick et al., 2015). In PPDB, each paraphrases pair $\{e_1, e_2\}$ is obtained from translation models making use of the observation that if e_1 and e_2 are frequently translated to a same word or phrase in a foreign language, then there is a high probability of e_1 and e_2 being paraphrases of each other. In PPDB 2.0 each pair is labeled with $-\log(P(e_1|e_2))$ and $-\log(P(e_2|e_1))$ obtained from the translation models, where $P(e_2|e_1)$ is the inferred probability that the word or phrase e_1 is a good paraphrase for e_2, and the contrary for $P(e_1|e_2)$. The motivation for using this resource is that text pairs that can be paraphrased to simplified versions should be easier to analyze by downstream modules. For example, consider the paraphrase pair $e_1 =$"*interdisciplinary*" and $e_2 =$"*cross-disciplinary*" labeled in PPDB with $-\log(P(e_2|e_1)) = 4.28$ and $-\log(P(e_1|e_2)) = 0.82$. Now, consider a pair of sentences for STS evaluation where these paraphrases occur: "The study was *interdisciplinary*." and "Our research is *cross-disciplinary*." Given that e_1 is higher scoring paraphrase for e_2 than the contrary (i.e. $-\log(P(e_2|e_1)) > -\log(P(e_1|e_2))$), e_2 can be replaced by e_1 in the second sentence: "Our research is *interdisciplinary*". As a result, the pair of sentences now contains more frequent words and shares more words thereby facilitating subsequent STS analysis.

Let A and B be a pair of texts snippets for STS evaluation and e_1 and e_2 a pair of paraphrases from PPDB. Thus, $\{e_1, e_2\}$ occurs in $\{A, B\}$ if $e_1 \subset A \wedge e_2 \subset B$ or $e_2 \subset A \wedge e_1 \subset B$, being aware of the special cases when $e_1 \subset e_2$ or $e_2 \subset e_1$(whole words matching is used in those cases). The operator "$\subset$" means that the left argument is a sub-string of the right one. The input pairs of sentences for the STS task where preprocessed by looking for occurrences of paraphrases from PPDB and replacing the least probable paraphrase by the most probable paraphrase. For that, we used the top-ranked lexical paraphrases and phrasal paraphrases from the M-size version of the PPDB 2.0 [1] (syntactic rules were not used). We determined the number of top-ranking lexical and phrasal paraphrases to use experimentally by using the overall STS system described in this paper trained and tested with STS datasets from previous years. Consistent increases in the performance measured by mean correlation was observed as the number of used paraphrases increased. The average relative improvement stabilized around 2% using 150,000 lexical paraphrases and 3,000,000 phrasal paraphrases. Using these thresholds for the paraphrase database we assessed the 14 thousand sentence pairs in training data and found 3,294 occurrences of lexical paraphrases and 1,778 phrasal paraphrases.

2.2 Tokenizing, Stop-words Removal and Negation Detection

The preprocessing continues by tokenizing sentences, removing stop-words and labeling negated

[1] http://paraphrase.org/#/download

words. For this stage, we use the tokenizer and stop-words list from NLTK[2] augmented with the following words: *should*, *now*, *'s*, *'t*, *'ve*, *something*, *would* and *also*. Once stop-words are removed from the text, each word preceded by a negation token is labeled as a negated word. The negation tokens we use are: *not*, *n't*, *nor*, *null*, *neither*, *barely*, *scarcely*, *hardly*, *no*, *none*, *nobody*, *nowhere*, *nothing*, *never* and *without*. The negation tagged tokens are used by subsequent modules for modeling oppositeness between negated and non-negated forms (e.g. "not running" and "running").

3 Lexical Similarity

The analysis of short texts based on soft cardinality relies only on a similarity function between lexical units (Jimenez et al., 2010). Therefore, the first component of the proposed STS system is composed of four lexical similarity functions that compare a pair of words and yield a numerical value in [-1,1] interval. Returning values of 1 means that the two words can be considered identical, 0 for unrelated words, negative values for representing opposition, and other values for representing intermediate degrees of similarity and opposition. In this section, the four lexical similarity functions we use are described.

3.1 Lexical String Matching Boosted with Synonyms and Antonyms

The NLP community has widely recognized that the use of lemmas or stems, instead of words, is desirable in many applications of automatic text processing. Therefore, before comparing any pair of words we reduced them to their stems using the Porter's algorithm (Porter, 1980). Let x and y be two stemmed words represented each as a sequence of characters. The first proposed lexical function replaces this basic word representation by the set of tri-grams and tetra-grams of characters on each word. This representation was used successfully for addressing the STS task with purely string-based approaches (Jimenez et al., 2012). For example, the word *country* is stemmed to $x =$*countri*. Next, its [3:4]-grams representation is $x = \{$*cou*, *oun*, *unt*, *ntr*, *tri*, *coun*, *ount*, *untr*, *ntri*$\}$. Once x and y are represented as described, they are compared with the following expression:

$$S_1(x,y) = \frac{|x \cap y|}{\sqrt{|x| \times |y|}}$$

The second lexical similarity function is the well-known Jaro-Winkler (Winkler, 1990) expression:

$$d(x,y) \quad = \quad \frac{1}{3}\left(\frac{m}{\text{len}(x)} + \frac{m}{\text{len}(y)} + \frac{m-t}{m}\right)$$

$$S_2 = \begin{cases} d(x,y) & \text{if } d(x,y) < b_t \\ d(x,y) + (lp \times (1 - d(x,y)) & \text{otherwise} \end{cases}$$

Where, $\text{len}(x)$ is the number of characters in word x, m is the number of matching characters between x and y, t is the number of transpositions between x and y, lp is the length of the common prefix, $p = 0.1$, and $b_t = 0.7$ is the "boost threshold". The number of matching characters m is the number of common characters between x and y whose occurrences are not farther than $\left\lfloor \frac{max[len(x), len(y)]}{2} \right\rfloor - 1$ positions. The number t of transpositions is the number of matching characters that occur in different sequence order on each string. Clearly, $m \leq \text{len}(x)$, $m \leq \text{len}(y)$, and $t \leq m$, therefore $d(x,y)$ is defined only in $[0,1]$ interval (if $m = 0$, then $d(x,y)$ is set to 0). Similarly to S_1, S_2 was used to compare stems instead of words.

Both S_1 and S_2 returns 1 when x and y are identical, 0 when x and y do not have common characters, and intermediate values for other cases. To a certain extent, this stem string similarity reflects semantic similarity between words. To improve this property, a wrapper function $\mathring{S}$ was built over S_1 and S_2 to include information from the synonym and antonym relationships in WordNet and the negation feature extracted at preprocessing stage (see subsection 2.2). If x and y are synonyms in WordNet, then the wrapper function $\mathring{S}$ overwrites the results of S_1 and S_2 to 1 (identical meaning). For the case when x and y are antonyms, the wrapper function should return a negative value to represent the opposition between words x and y (opposite meaning). Unlike synonymy and identity, the relation between antonymy and numerical oppositeness is rather unclear because most antonym pairs also are semantically similar (e.g. *small-large*) (Mohammad et al.,

[2] http://www.nltk.org/

2008). The natural choice for this negative values is -1 (Yih et al., 2012)(Yih et al., 2012). However, instead of setting -1 to represent oppositions between two words, we decided to set this value as a parameter to be determined experimentally. For that, we used the overall STS system described in this paper with the STS datasets from previous years. The value that optimized the mean correlation was -0.2 in a search range from -1 to 1. The negation feature of the words is used to add negation logic to $\mathring{S}$. For example, if x and y are synonyms but x is negated, then they are considered antonyms. Some examples are: $\mathring{S}(car, auto) = 1$, $\mathring{S}(\neg car, \neg auto) = 1$, $\mathring{S}(\neg car, auto) = -0.2$, $\mathring{S}(love, hate) = -0.2$, $\mathring{S}(\neg love, \neg hate) = -0.2$ and $\mathring{S}(\neg love, hate) = 1$ ($\neg$ signify "negated word"). In the remaining cases, when x and y are neither synonyms nor antonyms, the wrapper function returns $S_i(x, y)$ except for the case when either x or y is negated. In that later case, the wrapper function returns $0.26 \times S_i(x, y)$, which is a scaling factor for modeling negation determined experimentally in the same way as the opposite value of 0.2. A couple of examples are: $\mathring{S}_1(skater, skateboard) = 0.489$, $\mathring{S}_1(\neg skater, skateboard) = 0.489 \times 0.26 = 0.127$. Henceforth, functions S_1 and S_2 are assumed to be overwritten by the described wrapper function.

3.2 Word Embedding

Two additional lexical similarity functions were built using word embedding representations. Let $\vec{x}$ and $\vec{y}$ be the vectorial representations of words x and y in $\mathbb{R}^n$. The used lexical similarity function between a pair of words is the cosine between these vectorial representations:

$$S_3(x, y) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \times \|\vec{y}\|}$$

Function S_3 is computed using the publicly available pre-trained Google News corpus word embedding [3] from the *word2vec* tool (Mikolov, Tomas et al., 2013). We also include a similar function S_4 that is defined identically to S_3 but uses pre-trained Twitter corpus word embedding [4] from *the GloVe* tool (Pennington, Jeffrey et al., 2014). These cosine based similarities produce scores in the range from -1 to 1.

4 Polarized Soft Cardinality

Lexical similarity can be leveraged to address sentence similarity by aggregating lexical similarity scores. One successful mechanism for doing this is soft cardinality (Jimenez et al., 2010), which is a generalization of the classic set cardinality that considers similarities between elements. Thus, the soft cardinality of a bag of words $A = \{a_1, a_2, \ldots, a_n\}$ (i.e. a sentence) and a similarity function between words $S(a_i, a_j)$ is defined by this expression:

$$|A|_S = \sum_{i=1}^{n} \left(\frac{1}{\sum_{j=1}^{n} S(a_i, a_j)^p} \right)$$

Where p is the softness-control parameter, which is positive and its default value is $p = 1$. Soft cardinality is a generalization of classic cardinality because as p increases, $|A|_S$ gets closer to $|A|$. The soft cardinality of the union of two bags of words $|A \cup B|_S$ is simply the soft cardinality of the concatenation of the bags. The soft cardinality of the intersection of two bag is defined as $|A \cap B|_S = |A|_S + |B|_S - |A \cup B|_S$. This model is restricted only to positive lexical similarity functions because negative values could lead to division-by-zero if $\sum_{j=1}^{n} S(a_i, a_j)^p = 0$ for any a_i.

Given that the lexical similarity functions S_1 to S_4 (described in Section 3) can return negative values for words with opposite semantics, a new soft cardinality model that supports such negative similarities between elements was proposed for this competition. The new *polarized soft cardinality* model is:

$$\begin{aligned} |A|_S &= \sum_{i=1}^{n} \left(\frac{2 - \frac{1}{1 - \sum_{j=1}^{n} \mathrm{neg}(S(a_i, a_j), p)}}{\sum_{j=1}^{n} \mathrm{pos}(S(a_i, a_j), p)} \right) \\ \mathrm{neg}(s, p) &= \begin{cases} (-s)^p & \text{if } s < 0 \\ 0 & \text{otherwise} \end{cases} \\ \mathrm{pos}(s, p) &= \begin{cases} s^p & \text{if } s \geq 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Functions $\mathrm{neg}(s, p)$ and $\mathrm{pos}(s, p)$ filter respectively negative and positive values of s, then raise

[3] GoogleNews-vectors-negative300.bin downloaded from https://code.google.com/archive/p/word2vec/

[4] glove.twitter.27B.50d.txt downloaded from http://nlp.stanford.edu/projects/glove/

them to p power ignoring the sign. Note that, if $S(a_i, a_j)$ is strictly positive, then this model is equivalent to soft cardinality . This new model inserts dummy or "ghost" elements in A if there are opposite elements in A. For example, consider $A = auto, love, hate$ and $S(auto,\ love) = S(auto,\ hate) = S(love,\ hate) = 0$. Clearly, $|A|_S = 3$. However, if $S(love,\ hate) = -1$, then $|A|_S = 4$. This increment in soft cardinality reflects the presence of a dummy element due to the fact that *love* and *hate* are opposites.

Using the lexical similarity functions presented in Section 3, four soft cardinality functions can be built: $|*|_{S_1}$, $|*|_{S_2}$, $|*|_{S_3}$ and $|*|_{S_4}$. Each of those has the following softness control parameter values: $p_{S_1} = 1.05$, $p_{S_2} = 0.85$, $p_{S_3} = 0.5$ and $p_{S_4} = 0.65$, which were obtained experimentally using STS data from previous SemEval campaigns. These soft cardinality functions are used to extract numerical features from each pair of sentences to be evaluated (see Section 6).

5 L^2-norm Cardinality

Given two sentences A and B represented as bag of words, the proposed *L^2-norm cardinality* is a measure of the amount of information in A, B, $A \cap B$ and $A \cup B$. L2-norm cardinality is analogous to soft cardinality but uses vectorial representations of the words and vector operations in its formulation. Instead of exploiting pairwise similarities between words as soft cardinality does, L^2-norm cardinality uses vectorial representations of the words in the "bag" to assess its cardinality. Let $A = \{\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_p\}$ and $B = \{\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_q\}$ be a bag of vector-represented words in $\mathbb{R}^n$, where p and q are the number of words in A and B respectively. Firstly, A and B obtain a representation in $\mathbb{R}^n$ by adding up the vectors in their respective bags, i.e. $\mathbf{A} = \sum_{i=1}^{p} \vec{a}_i$ and $\mathbf{B} = \sum_{i=1}^{q} \vec{b}_i$. L^2-norm cardinality is defined by the following expressions:

$$\begin{aligned} |A|_n &= \|\mathbf{A}\|^2 \\ |B|_n &= \|\mathbf{B}\|^2 \\ |A \cap B|_n &= \mathbf{A} \cdot \mathbf{B} \\ |A \cup B|_n &= |A|_n + |B|_n - |A \cap B|_n \end{aligned}$$

Two L^2-norm cardinality functions can be built reusing the same word embedding used in S_3 and S_4. Thus, $|*|_{300}$ is obtained using the pre-trained word2vec vectors and $|*|_{50}$ is obtained from the pre-trained GloVe vectors. L^2-norm cardinalities $|*|_{300}$ and $|*|_{50}$ are added to the set of the four previously proposed soft cardinality functions to be used for extracting numerical features from sentence pairs.

6 Feature Extraction

The six cardinality functions proposed in Section 4 and Section 5 can be used to build a variety of similarity assessment measures for STS.. For example, for sentences A and B, the expression $sim(A, B) = \frac{|A \cap B|_{S_1}}{|A \cup B|_{S_1}}$ is a possible STS measure based on Jaccard's coefficient. However, the space of possible similarity functions that can be built from cardinalities $|A|_{S_1}$, $|B|_{S_1}$, $|A \cup B|_{S_1}$ and $|A \cap B|_{S_1}$ is huge. We explore a limited portion of this space by generating similarity function expressions from a set of 11 factors (see Table 1). Parameter c in Table 1 represents the sub-index for identifying the possible cardinality function, $c \in \{S_1, S_2, S_3, S_4, 300, 50\}$. The set of expressions used for combining these factors is heuristic but motivated by the formulations of existing cardinality-based similarity measures (e.g. Jaccard, Dice, matching, cosine among others). For each one of the six cardinality functions, these factors were combined into expressions that generate a total of $11 \times 10 = 110$ features of the form $\frac{f_i}{f_j}$; $i \neq j$.

7 Feature Selection

The feature selection process consists in selecting the k-best features from the set of 110 features for each cardinality function. We used the method *SelectKBest*[5] from the *Scikit-learn* machine learning kit (Pedregosa, Fabian et al., 2011). The data used for this selection process was the concatenation of 20 STS datasets labeled with gold standard from the past SemEval campaigns from the years 2012 to 2015 (14,437 sentence pairs with gold standard annotations). The process was performed using 10-fold cross-validation repeating the selection ten times with different randomly selected fold partitions. The k features that were selected the most

[5] http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

Factor	Expression†	Name
f_1	$\lvert A \cap B \rvert_c$	intersection
f_2	$\lvert A \cup B \rvert_c$	union
f_3	$\lvert A \triangle B \rvert_c$	symmetric diff.
f_4	$\min[\lvert A \rvert_c, \lvert B \rvert_c]$	minimum
f_5	$\max[\lvert A \rvert_c, \lvert B \rvert_c]$	maximum
f_6	$0.5 \times (\lvert A \rvert_c + \lvert B \rvert_c)$	mean
f_7	$\sqrt{\lvert A \rvert_c \times \lvert B \rvert_c}$	geometric mean
f_8	$\sqrt{0.5 \times (\lvert A \rvert_c^2 + \lvert B \rvert_c^2)}$	quadratic mean
f_9	$\sqrt[3]{0.5 \times (\lvert A \rvert_c^3 + \lvert B \rvert_c^3)}$	cubic mean
f_{10}	$\sqrt[4]{0.5 \times (\lvert A \rvert_c^4 + \lvert B \rvert_c^4)}$	4^{th}mean
f_{11}	1.0	1.0

† $c \in \{S_1, S_2, S_3, S_4, 300, 50\}$

Table 1: Factors for rational similarity functions based on cardinality functions

times in the k-best selection after all runs were retained for the final model. Preliminary experiments suggest a good value for k is two, as determined using the overall STS system with the same data. Table 2 shows the selected features for each cardinality function and their results on the mean correlation performance measure as assessed under previous STS shared task evaluation settings. Although, none of the features outperformed the best official results, results of $| * |_{S_1}$ and $| * |_{300}$ cardinality functions are highly competitive. It is important to note that, in spite that the feature selection procedure is supervised, the selected features by themselves are inherently un-supervised.

8 Feature Combination

The 12 selected features showed in Table 2 were combined to produce predictions for our three participating runs. *Run1* and *run2* were SVR (support-vector regression) models with RBF kernel (Drucker, Harris et al., 1997) built with the 14,437 sentence pairs available for training. The difference between *run1* and *run2* is the values of the used SVR parameters C and γ. For *run1*, we used $C = 0.6$ and $\gamma = 0.004$, which were obtained by optimizing the weighted average of Pearson correlations using each available STS dataset alternatively for testing and the remaining pairs for training. For *run2*, we used $C = 53$ and $\gamma = 0.012$, which were obtained using all 14,437 sentence pairs as a single dataset and 5-folds cross-validation in 5 randomly selected division folds.

Unlike *run1* and *run2*, *run3* was effectively unsupervised and operated by simply averaging the 12 feature values after multiplying by -1 the values of the features with negative correlations in Table 2.

9 Cross-lingual Sub-task

The predictions of the three runs submitted to the STS cross-lingual sub-task were produced using the same systems that produced predictions for the STS monolingual sub-task (English). For that, the texts in Spanish were translated into English using Google's public translate service.[6]

10 Results

Table 3 shows results obtained with the same systems used to produce our runs 1, 2 and 3, but using datasets from STS competitions from 2012 to 2015. The results labeled as "held-out" were obtained by holding out each dataset for testing and using the remaining datasets for training including the three training datasets from 2012. The results labeled as "same-data" were otained using the same training data available during each historical STS evaluation. All "held-out" systems outperformed consistently the best official results obtained by a single system for each year using the performance measure of weighted mean correlation. For the "same-data" systems, the *run1* outperformed historical official results from 2013 to 2015, while *run2* and *run3* made it only for 2013 and 2014. Table 4 shows results of the same systems ("held-out" testing setting) and the best official results obtained on each dataset of the 20 individual evaluation datasets from prior STS competitions. In this tougher comparison, the proposed systems obtained state-of-the-art results in 10 out of the 20 individual datasets, and competitive results for the majority of the remaining datasets. Finally, Table 5 and Table 6 show the results obtained by our systems on the 2016 datasets along with the best official results of the competition. When comparing our three runs, none of them was consistently better with the three runs typically obtaining similar results. Therefore, it is possible to conclude that the

[6] https://translate.google.com/

Cardinality function	Feature	Expression	2012	2013	2014	2015
$\vert * \vert_{S_1}$ SC+[3:4]grams	f_3/f_8	$\frac{\vert A \triangle B\vert_{S_1}}{\sqrt{0.5\times(\vert A\vert^2_{S_1}+\vert B\vert^2_{S_1})}}$	**-0.6404**	-0.5994	-0.7538	**-0.7953**
	f_3/f_9	$\frac{\vert A \triangle B\vert_{S_1}}{\sqrt[3]{0.5\times(\vert A\vert^3_{S_1}+\vert B\vert^3_{S_1})}}$	-0.6376	-0.5952	**-0.7534**	-0.7933
$\vert * \vert_{S_2}$ SC+Jaro-Winkler	f_1/f_2	$\frac{\vert A\cap B\vert_{S_2}}{\vert A\cup B\vert_{S_2}}$	0.4327	0.3287	0.3793	0.4012
	f_3/f_2	$\frac{\vert A\triangle B\vert_{S_2}}{\vert A\cup B\vert_{S_2}}$	-0.4327	-0.3385	-0.3793	-0.4012
$\vert * \vert_{S_3}$ SC+word2vec	f_8/f_2	$\frac{\sqrt{0.5\times(\vert A\vert^2_{S_3}+\vert B\vert^2_{S_3})}}{\vert A\cup B\vert_{S_3}}$	0.5671	0.4722	0.5572	0.5831
	f_9/f_2	$\frac{\sqrt[3]{0.5\times(\vert A\vert^3_{S_3}+\vert B\vert^3_{S_3})}}{\vert A\cup B\vert_{S_3}}$	0.5652	0.4702	0.5604	0.5804
$\vert * \vert_{S_4}$ SC+GloVe	f_{11}/f_1	$\frac{1.0}{\vert A\cap B\vert_{S_4}}$	-0.2195	-0.2331	-0.3014	-0.2961
	f_3/f_2	$\frac{\vert A\triangle B\vert_{S_4}}{\vert A\cup B\vert_{S_4}}$	-0.4348	-0.2915	-0.3168	-0.3093
$\vert * \vert_{300}$ L^2-norm+word2vec	f_3/f_{11}	$\frac{\vert A\triangle B\vert_{300}}{1.0}$	-0.6389	**-0.6065**	-0.7334	-0.7509
	$f_{3,}/f_5$	$\frac{\vert A\triangle B\vert_{300}}{\max[\vert A\vert_{300},\vert B\vert_{300}]}$	-0.6389	-0.6022	-0.7334	-0.7509
$\vert * \vert_{50}$ L^2-norm+GloVe	f_1/f_2	$\frac{\vert A\cap B\vert_{50}}{\vert A\cup B\vert_{50}}$	0.5377	0.4750	0.5808	0.5744
	f_3/f_2	$\frac{\vert A\triangle B\vert_{50}}{\vert A\cup B\vert_{50}}$	-0.5377	-0.4750	-0.5808	-0.5744
Best official result at SemEval			**0.6773**	**0.6181**	**0.7610**	**0.8015**

Table 2: Results of mean correlation (official performance measure) of the 2-best features for each cardinality function in previous years STS data

System	2012	2013	2014	2015
run1.held-out	**0.6951**	**0.6393**	0.7842	**0.8101**
run2.held-out	0.6895	0.6367	0.7826	0.8013
run3.held-out	0.6945	0.6383	**0.7851**	0.8099
run1.same-data	**0.6771**	0.6322	**0.7692**	**0.8048**
run2.same-data	0.6696	0.6241	0.7677	0.7931
run3.same-data	0.6721	**0.6327**	0.7553	0.7925
best official result	0.6773	0.6181	0.7610	0.8015

Table 3: Results of our 2016 systems using data from previous STS SemEval competitions (mean correlation performance measure).

contribution of SVR was not considerable, with an exception for the *answer-answer* dataset.

11 Conclusion

The proposed STS system combined effectively the most popular resources used by the top systems during the SemEval 2015 for STS shared task. Results show that the proposed system outperformed all past systems in a per-system based comparison, and obtained state-of-the-art results in half of the datasets from past STS competitions at SemEval.

Yr.	Dataset	*run1*	*run2*	*run3*	best†
2012	MSRpar	0.6522	0.6549	0.5829	**0.7343**
	MSRvid	0.8520	0.8612	0.8494	**0.8803**
	SMTeurop.	0.5332	0.5285	0.5499	**0.5666**
	OnWN	0.7270	0.7120	0.7228	**0.7273**
	SMTnews	0.6068	0.5750	0.5965	**0.6085**
2013	FNWN	0.4705	0.3920	0.4721	**0.5818**
	headlines	**0.8006**	0.8020	0.7836	0.7838
	OnWN	0.7865	0.8057	0.7562	**0.8431**
	SMT	0.4105	0.4065	**0.4165**	0.4035
	deft-forum	**0.5512**	0.5307	0.5464	0.5305
	deft-news	**0.7925**	0.7757	0.7823	0.7850
2014	headlines	**0.7913**	0.7914	0.7768	0.7837
	OnWN	0.8367	0.8388	0.8224	**0.8745**
	tweet-news	0.8019	0.8027	**0.8148**	0.7921
	images	**0.8435**	0.8514	0.8324	0.8343
2015	ans.-forums	**0.7474**	0.7066	0.7364	0.7390
	ans.-studt.	0.7853	0.7698	**0.7900**	0.7879
	belief	0.7536	0.7464	0.7496	**0.7717**
	headlines	0.8307	0.8289	0.8183	**0.8417**
	images	0.8740	**0.8797**	0.8712	0.8713

†Official results of the best system for each dataset

Table 4: Results of our 2016 systems being compared against best official results in a comparison for each dataset.

2016 dataset	*run1*	*run2*	*run3*	best
answer-answer	0.5018	0.5526	0.4907	0.6924
headlines	0.7865	0.7830	0.7773	0.8275
plagiarism	0.8365	0.8151	0.8293	0.8414
postediting	0.8364	0.8163	0.8481	0.8669
question-quest.	0.6652	0.6663	0.6729	0.7471
ALL mean r	0.7241	0.7262	0.7222	0.7781†

†Best individual system submission

Table 5: Official results for our participating systems in the monolingual sub-task (English).

2016 dataset	*run1*	*run2*	*run3*	best
News	0.8872	0.8291	0.8965	0.9124
Multisource	0.8184	0.8127	0.8074	0.8190
ALL mean r	0.8532	0.8210	0.8525	0.8631
Run Rank	3^{rd}	7^{th}	4^{th}	1^{st}
Team Rank	2^{nd}	2^{nd}	2^{nd}	1^{st}

Table 6: Official results for our participating systems in the cross-lingual sub-task (English/Spanish).

Acknowledgments

I thank the anonymous reviewers and Daniel Cer for their constructive comments. In addition, I am indebted to my beloved Claudia Luisa, and Sara, who kindly conceded part of our time together for this participation.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Gonzalez-Agirre Aitor. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In *First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 385–393, Montreal,Canada. ACL.

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. pages 32–43, Atlanta, Georgia, USA. ACL.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 Task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. ACL.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, and others. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263. ACL.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2016. SemEval-2016 Task 1: Semantic Textual Similarity - Monolingual and Cross-lingual Evaluation. ACL.

Drucker, Harris, Burges, Chris J.C., Kaufman, Linda, Smola, Alex, and Vapnik, Vladimir. 1997. Support vector regression machines. *Advances in neural information processing systems*, 9:155–161.

Sergio Jimenez, Fabio Gonzalez, and Alexander Gelbukh. 2010. Text Comparison Using Soft Cardinality. In Edgar Chavez and Stefano Lonardi, editors, *String Processing and Information Retrieval*, volume 6393 of *LNCS*, pages 297–302. Springer, Berlin, Heidelberg.

Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. 2012. Soft Cardinality: A Parameterized Similarity Function for Text Comparison. In *First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 449–453, Montreal, Canada. ACL.

Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. 2013a. SOFTCARDINALITY-CORE: Improving Text Overlap with Distributional Measures for Semantic Textual Similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task*, pages 194–201, Atlanta, Georgia, USA, June. ACL.

Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. 2013b. SOFTCARDINALITY: Learning to Identify Directional Cross-Lingual Entailment from Cardinalities and SMT. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 34–38, Atlanta, Georgia, USA, June. ACL.

Sergio Jimenez, George Duenas, Julia Baquero, and Alexander Gelbukh. 2014. UNAL-NLP: Combining soft cardinality features for semantic textual similarity, relatedness and entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 732–742, Dublin, Ireland. ACL.

André Lynum, Partha Pakray, Björn Gambäck, and Sergio Jimenez. 2014. NTNU: Measuring Semantic Similarity with Sublexical Feature Representations and Soft Cardinality. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 448–453, Dublin, Ireland. ACL.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg, and Dean, Jeff. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119.

Miller, George A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

Saif Mohammad, Bonnie Dorr, and Graeme Hirst. 2008. Computing word-pair antonymy. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 982–991. Association for Computational Linguistics.

Ellie Pavlick, Johan Bos, Malvina Nissim, Charley Beller, Ben Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proc. ACL*.

Pedregosa, Fabian, Varoquaux, Gaël, Gramfort, Alexandre, Michel, Vincent, Thirion, Bertrand, Grisel, Olivier, Blondel, Mathieu, Prettenhofer, Peter, Weiss, Ron, Doubourg, Vincent, Vanderplas, Jake, Passos, Alexandre, Cournapeau, David, Brucher, Matthieu, Perrot, Matthieu, and Duchesnay, Edouard. 2011. Scikit-learn: Machine Learning in {P}ython. *The Journal of Machine Learning Research*, 12:2825–2830.

Pennington, Jeffrey, Socher, Richard, and Manning, Christopher D. 2014. Glove: Global vectors for word representation. In *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, volume 12, pages 1532–1543, Doha, Qatar.

Martin Porter. 1980. An algorithm for suffix stripping. *Program*, 3(14):130–137, October.

William E. Winkler. 1990. String comparator metrics and enhanced decision Rules in the Fellegi-Sunter Model of Record Linkage. In *Proceedings of the Section on Survey Research Methods*, pages 354–359. American Statistical Association.

Wen-tau Yih, Geoffrey Zweig, and John C. Platt. 2012. Polarity inducing latent semantic analysis. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1212–1222. Association for Computational Linguistics.