Soft Cardinality-CORE: Improving Text Overlap with Distributional Measures for Semantic-Textual Similarity

Sergio Jimenez
Universidad Nacional
de Colombia
Ciudad Universitaria,
edificio 453, oficina 114
Bogotá, Colombia
[sgjimenezv|cjbecerrac]

Claudia Becerra
Universidad Nacional
de Colombia
Ciudad Universitaria,
edificio 453, oficina 114
Bogotá, Colombia
@unal.edu.co

Alexander Gelbukh
CIC-IPN
Av. Juan Dios Bátiz, Av. Mendizábal,
Col. Nueva Industrial Vallejo
CP 07738, DF, México
gelbukh@gelbukh.com

Abstract

The soft cardinality proved to be a very strong text-overlapping baseline for the task of semantic-textual-similarity (STS) obtaining the third place in SemEval-2012. This year, besides to the plain text-overlapping approach, two distributional word-similarity functions derived from the ukWack corpus were tested within the soft cardinality. These measures contributed to improve the performance of the text-overlapping approach. Further, these were combined with other features using regression obtaining positions 18th, 22th and 23th among the 90 participants systems in the official 2013 shared task ranking at *SEM. After the release of the gold standard anotations of the test data, we observed that the bare similarity measures, without the use of regression, would have obtained positions 6th, 7th and 8th. Moreover, the simple arithmetic average of these similarity measures would have been 4th (*mean*=0.5747). This paper describes the submitted system and the similarity measures that would obtained those better results.

1 Introduction

The task of textual semantic similarity (STS) consists in providing a similarity function that takes pairs of texts as arguments and returns a result that correlates with human judgments. This function has many practical applications in NLP tasks (e.g. summarization, question answering, textual entailment, paraphrasing, machine translation evaluation, etc.) making it a task of special interest. Although several previous efforts have been made in the investigation of this central problem (Lee et al., 2005; Mihalcea et al., 2006), major campaigns were held in SemEval 2012 (Agirre et al., 2012) and in *SEM 2013 (Agirre et al., 2013).

The experimental setup of STS in 2012 consisted of three data sets, roughly divided into 50% for training and test, containing text pairs annotated with a gold standard. Furthermore, two data sets were provided for surprise testing. The measure of performance was the average of the correlations per data set weighted by the number of pairs in each data set (*mean*). The best performing systems were UKP (Bär et al., 2012) *mean*=0.6773, TakeLab (Šaric et al., 2012) *mean*=0.6753 and soft cardinality (Jimenez et al., 2012) *mean*=0.6708. UKP and TakeLab systems used a long list of resources (see (Agirre et al., 2012)) including dictionaries, distributional thesaurus, monolingual corpora, Wikipedia, WordNet, distributional similarity, KB similarity, POS tagger, machine learning and more. Unlike these systems, the soft cardinality approach used mainly text overlapping and common-practice text preprocessing such as the elimination of stop words, stemming and *idf* term weighting. From this result, we may conclude that the additional gain in performance due to the use of external resources is small and that the soft cardinality approach is a challenging baseline for the STS task. Jimenez and Gelbukh (2012) showed that the soft cardinality is also a good baseline for other applications such as information retrieval, entity matching, paraphrase detection and textual entailment.

The soft cardinality (Jimenez et al., 2010) is an approach for the construction of similarity functions

using any cardinality-based resemblance coefficient (e.g. Jaccard or Dice) but substituting the classical-set-cardinality by a softened counting function. For example, the soft cardinality of a set containing three almost-identical elements is close to (but larger than) 1 and for three hardly-distinguishable elements is close to (but less than) 3. To use the soft cardinality with texts, these must be represented as sets of words and a word-similarity function is required to do the soft counting of the words. For the sake of completeness, a brief recap of the soft-cardinality method is provided in Section 3.

The resemblance coefficient used in our participation is a modified version of the Tversky's ratio model (Tversky, 1977). Apart from the two parameters of this coefficient, a new parameter was included and functions max/min were used to do it symmetrical. The rationale for this new coefficient is presented in Section 2.

Three word similarity features used in our systems are described in Section 2.4. The first of them, is a measure of character q-grams overlapping, which reuses the coefficient proposed in Section 2; this measure is described in subsection 4.1. The second and third are distributional measures obtained from the ukWack corpus (Baroni et al., 2009), which is a collection of web-crawled documents containing about 1.9 billion words in English. The second measure is again a reuse of the coefficient specified in Section 2, but using sets of occurrences (and co-occurrence) of words in sentences in the ukWack corpus; this measure is described in subsection 4.2. Finally the third, which is a normalized version of pointwise mutual information (PMI), is described in subsection 4.3.

The parameters of the three text-similarity functions derived from the combination of the proposed coefficient of resemblance (section 2), the soft cardinality (section 3) and the three word-similarity measures (section 4) were adjusted to maximize the correlation with the gold standard in 2012 STS data. At this point, these soft-cardinality similarity functions can provide predictions for the test data. However, we decided to test the approach of learning from the training data a resemblance function instead of using a preset resemblance coefficient. Basically, most resemblance coefficients are ternary functions $F(x, y, z)$ where $x = |A|$, $y = |B|$ and $z = |A \cap B|$ e.g. Dice's coefficient is $F(x, y, z) = {}^{2z}/_{x+y}$ and Jaccard's is $F(x, y, z) = {}^{z}/_{x+y-z}$. Thus, this function can be learned using a regression model, providing cardinalities x, y and z as features and the gold standard as target function. The results obtained by the text-similarity functions and the regression approach are presented in Section 7.

Unfortunately, when using a regressor trained with 2012 STS data and tested with 2013 surprise data we observed results worsened rather than improved. The short explanation of this phenomenon is overfitting. A more constructive discussion of this result, together with an assessment of the performance gain obtained by the use of distributional measures is provided in Section 8.

Finally, in Section 9 the conclusions of our participation in this evaluation campaign are presented.

2 Symmetrical Tversky's Ratio Model

In the field of mathematical psychology Tversky proposed the ratio model (TRM) (Tversky, 1977) motivated by the imbalance that humans have on the selection of the referent to compare things. This model is a parameterized resemblance coefficient to compare two sets A and B given by the following expression:

$$\mathbf{trm}(A, B) = \frac{|A \cap B|}{\alpha|A - B| + \beta|B - A| + |A \cap B|},$$

Having $\alpha, \beta \geq 0$. The numerator represents the commonality between A and B, and the denominator represents the referent for comparison. Parameters α and β represent the preference in the selection of A or B as referent. Tversky associated the set cardinality, to the stimuli of the objects being compared. Let's consider a Tversky's example of the 70s: A is North Corea, B is red China and stimuli is the prominence of the country. When subjects assessed the similarity between A and B, they tended to select the country with less prominence as referent. Tversky observed that α was larger than β when subjects compared countries, symbols, texts and sounds. Our motivation is to use this model by adjusting the parameters α and β for better modeling human similarity judgments for short texts.

However, this is not a symmetric model and the parameters α and β, have the dual interpretation of

modeling the asymmetry in the referent selection, while controlling the balance between $|A \cap B|$ and $|A - B| + |B - A|$ as well. The following reformulation, called symmetric TRM (**strm**), is intended to address these issues:

$$\mathbf{strm}(A, B) = \frac{c}{\beta\left(\alpha a + (1-\alpha)\, b\right) + c}, \qquad (1)$$

$a = \min(|A - B|, |B - A|)$, $b = \max(|A - B|, |B - A|)$ and $c = |A \cap B| + bias$. In $\mathbf{strm}$, α models only the balance between the differences in the cardinalities of A and B, and β models the balance between $|A \cap B|$ and $|A - B| + |B - A|$. Furthermore, the use of functions $\min$ and $\max$ makes the measure to be symmetric. Although the motivation for the $bias$ parameter is empirical, we believe that this reduces the effect of the common features that are frequent and therefore less informative, e.g. stop words. Note that for $\alpha = 0.5$,$\beta = 1$ and $bias = 0$, $\mathbf{strm}$ is equivalent to the Dice's coefficient. Similarity, for $\alpha = 0.5$,$\beta = 2$ and $bias = 0$, $\mathbf{strm}$ is equivalent to the Jaccard's coefficient.

3 Soft Cardinality

The cardinality of a set is its number of elements. By definition, the sets do not allow repeated elements, so if a collection of elements contains repetitions its cardinality is the number of different elements. The classical set cardinality does not take into account similar elements, i.e. only the identical elements in a collection counted once. The soft cardinality (Jimenez et al., 2010) considers not only identical elements but also similar using an auxiliary similarity function **sim**, which compares pairs of elements. This cardinality can be calculated for a collection of elements A with the following expression:

$$|A|' = \sum_{i=1}^{n} w_i \left(\sum_{j=1}^{n} \mathbf{sim}(a_i, a_j)^p \right)^{-1} \qquad (2)$$

$A = \{a_1, a_2, \ldots, a_n\}$; $w_i \geq 0$; $p \geq 0$; $1 \geq \mathbf{sim}(x, y) \geq 0$, $x \neq y$; and $\mathbf{sim}(x, x) = 1$. The parameter p controls the degree of "softness" of the cardinality. This formulation has the property of reproducing classical cardinality when p is large and/or when **sim** is a rigid function that returns 1 only for identical elements and 0 otherwise. The coefficients w_i are the weights associated with each element. In text applications elements a_i are words and weights w_i represent the importance or informative character of each word (e.g. *idf* weights).

4 Word Similarity

Analogous to the STS, the word similarity is the task of measuring the relationship of a couple of words in a way correlated with human judgments. Since when Rubenstein and Goodenough (1965) provided the first data set, this task has been addressed primarily through semantic networks (Resnik, 1999; Pedersen et al., 2004) and distributional measures (Agirre et al., 2009). However, other simpler approaches such as edit-distance (Levenshtein, 1966) and stemming (Porter, 1980) can also be used. For instance, the former identifies the similarity between "song" and "sing", and later that between "sing" and "singing". This section presents three approaches for word similarity that can be plugged into the soft cardinality expression in eq. 2.

4.1 *Q*-grams similarity

Q-grams are the collection of consecutive-overlapped substrings of length *q* obtained from the character string in a word. For instance, the 2-grams (bigrams) and 3-grams (trigrams) representation of the word "sing" are {'#s', 'si', 'in', 'ng', 'g#'} and {'#si', 'sin', 'ing', 'ng#'} respectively. The character '#' is a padding character that distinguishes *q*-grams at the beginning and ending of a word. If the number of characters in a word is greater or equal than q its representation in *q*-grams is the word itself (e.g. the 6-grams in "sing" are {'sing'}). Moreover, the 1-grams (unigrams) and 0-grams representations of "sing" are {'s', 'i', 'n', 'g'} and {'sing'}. A word can also be represented by combining multiple representations of *q*-grams. For instance, the combined representation of "sing" using 0-grams, unigrams, and bigrams is {'sing', 's', 'i', 'n', 'g', '#s', 'si', 'in', 'ng', 'g#'}, denoted by [0:2]-grams. In practice a range $[q_1 : q_2]$ of *q*-grams can be used having $0 \leq q_1 < q_2$.

The proposed word-similarity function (named $\mathbf{qgrams}$) first represents a pair of words using $[q_1 : q_2]$-grams and then compares them reusing

the $\mathbf{strm}$ coefficient (eq.1). The parameters of the $\mathbf{qgrams}$ function are q_1, q_2, α_{qgrams}, β_{qgrams}, and $bias_{qgrams}$. These parameters are sub-scripted to distinguish them from their counterparts at the text-similarity functions.

4.2 Context-Set Distributional Similarity

The hypothesis of this measure is that the co-occurrence of two words in a sentence is a hint of the possible relationship between them. Let's define $\mathrm{sf}(t)$ as the sentence frequency of a word *t* in a corpus. Similarly $\mathrm{sf}(t_A \wedge t_B)$ is the number of sentences where words t_A and t_B co-occur. The idea is to compute a similarity function between t_A and t_B representing them as A and B, which are sets of the sentences where t_A and t_B occur. Similarity, $A \cap B$ is the set of sentences where both words co-occur. The required cardinalities can be obtained from the sentence frequencies by: $|A| = \mathrm{sf}(t_A)$; $|B| = \mathrm{sf}(t_B)$ and $|A \cap B| = \mathrm{sf}(t_A \wedge t_B)$. These cardinalities are combined reusing again the $\mathbf{strm}$ coefficient (eq. 1) to obtain a word-similarity function. The parameters of this function, which we refer to it as **csds**, are α_{csds}, β_{csds} and $bias_{csds}$.

4.3 Normalized Point-wise Mutual Information

The pointwise mutual information (PMI) is a measure of relationship between two random variables. PMI is calculated by the following expression:

$$\mathbf{pmi}(t_A, t_B) = \log_2 \left(\frac{P(t_A \wedge t_B)}{P(t_A) \cdot P(t_B)} \right)$$

PMI has been used to measure the relatedness of pairs of words using the number of the hits returned by a search engine (Turney, 2001; Bollegala et al., 2007). However, PMI cannot be used directly as $\mathbf{sim}$ function in eq.2. The alternative is to normalize it dividing it by $\log_2(P(t_A \wedge t_B))$ obtaining a value in the $[1, -1]$ interval. This measure returns 1 for complete co-occurrence, 0 for independence and -1 for "never" co-occurring. Given that the results in the interval (0,-1] are not relevant, the final normalized-trimmed expression is:

$$\mathbf{npmi}(t_A, t_B) = \max \left[\frac{\mathbf{pmi}(t_A, t_B)}{\log_2(P(t_A \wedge t_B))}, 0 \right] \quad (3)$$

The probabilities required by PMI can be obtained by MLE using sentence frequencies in a large corpus: $P(t_A) \approx \frac{\mathrm{sf}(t_A)}{S}$, $P(t_B) \approx \frac{\mathrm{sf}(t_B)}{S}$,and $P(t_A \wedge t_B) \approx \frac{\mathrm{sf}(t_A \wedge t_B)}{S}$. Where S is the total number of sentences in the corpus.

5 Text-similarity Functions

The "building blocks" proposed in sections 2, 3 and 4, are assembled to build three text-similarity functions, namely $\mathbf{STS_{qgrams}}$, $\mathbf{STS_{csds}}$ and $\mathbf{STS_{npmi}}$. The first component is the $\mathbf{strm}$ resemblance coefficient (eq. 1), which takes as arguments a pair of texts represented as bags of words with importance weights associated with each word. In the following subsection 5.1 a detailed description of the procedure for obtaining such weighted bag-of-words is provided.

The $\mathbf{strm}$ coefficient is enhanced by replacing the classical cardinality by the soft cardinality, which exploits two resources: importance weights associated with each word (weights w_i) and pairwise comparisons among words (**sim**). Unlike $\mathbf{STS_{qgrams}}$ measure, $\mathbf{STS_{csds}}$ and $\mathbf{STS_{npmi}}$ measures require statistics from a large corpus. A brief description of the used corpus and the method for obtaining such statistics is described in subsection 5.2. Finally, the three proposed text-similarity functions contain free parameters that need to be adjusted. The method used to get those parameters is described subsection 5.3.

5.1 Preprocessing and Term Weighting

All training and test texts were preprocessed with the following sequence of actions: *i*) text strings were tokenized, *ii*) uppercase characters are converted into lowercase equivalents, *iii*) stop-words were removed, *iv*) punctuation marks were removed, and *v*) words were stemmed using Porter's algorithm (1980). Then each stemmed word was labeled with their *idf* weight (Jones, 2004) calculated using the entire collection of texts.

5.2 Sentence Frequencies from Corpus

The sentence frequencies $\mathrm{sf}(t)$ and $\mathrm{sf}(t_A \wedge t_B)$ required by $\mathbf{csds}$ and $\mathbf{npmi}$ word-similarity functions were obtained from the ukWack corpus (Baroni et al., 2009). This corpus have roughly 1.9 billion

words, 87.8 millions of sentences and 2.7 millions of documents. The corpus was iterated sentence by sentence with the same preprocessing that was described in the previous section, looking for all occurrences of words and word pairs from the full training and test texts. The target words were stored in a trie, making the entire corpus iteration took about 90 minutes in a laptop with 4GB and a 1.3Ghz processor.

5.3 Parameter optimization

The three proposed text-similarity functions have several parameters: p exponent in the soft cardinality; α, β, and $bias$ in **strm** coefficient; their sub-scripted versions in **qgrams** and **csds** word-similarity functions; and finally $q_1$and q_2 for **qgrams** function. Parameter sets for each of the three text-similarity functions were optimized using the full STS-SemEval-2012 data. The function to maximize was the correlation between similarity scores against the gold standard in the training data. The set of parameters for each similarity function were optimized using a greedy hill-climbing approach by using steps of 0.01 for all parameters except q_1 and q_2 that used 1 as step. The initial values were $p = 1$, $\alpha = 0.5$, $\beta = 1$, $bias = 0$, $q_1 = 2$ and $q_2 = 3$. All parameters were optimized until improvement in the function to maximize was below 0.0001. The obtained values are :

$\mathbf{STS_{qgrams}}$ $p = 1.32$,$\alpha = 0.52$, $\beta = 0.64$, $bias = -0.45$, $q_1 = 0$, $q_2 = 2$, $\alpha_{qgrams} = 0.95$, $\beta_{qgrams} = 1.44$, $bias_{qgrams} = -0.44$.

$\mathbf{STS_{csds}}$ $p = 0.5$, $\alpha = 0.63$, $\beta = 0.69$, $bias = -2.05$, $\alpha_{csds} = 1.34$, $\beta_{csds} = 2.57$, $bias_{csds} = -1.22$.

$\mathbf{STS_{npmi}}$ $p = 6.17$,$\alpha = 0.83$, $\beta = 0.64$, $bias = -2.11$.

6 Regression for STS

The use of regression is motivated by the following experiment. First, a synthetic data set with 1,000 instances was generated with the following three features: $|A| = RandomBetween(1, 100)$, $|B| = RandomBetween(1, 100)$ and $|A \cap B| = RandomBetween(0, min[|A|, |B|])$. Secondly, a linear regressor was trained using the Dice's coefficient (i.e. $2|A \cap B|/|A| + |B|$) as target function. The Pearson correlation obtained using 4-fold cross-validation as method of evaluation was $r = 0.93$. Besides, a Reduced Error Pruning (REP) tree (Witten and Frank, 2005) boosted with 30 iterations of Bagging (Breiman, 1996) was used instead of the linear regressor obtaining $r = 0.99$. We concluded that a particular resemblance coefficient can be accurately approximated using a nonlinear regression algorithm and training data.

#1	$\mathbf{STS_{sim}}$	#11	${}^{\lvert A\cap B\rvert'}/_{\lvert A\rvert'}$
#2	$\lvert A\rvert'$	#12	${}^{\lvert A\cap B\rvert'}/_{\lvert B\rvert'}$
#3	$\lvert B\rvert'$	#13	$\lvert A\rvert' \cdot \lvert B\rvert'$
#4	$\lvert A \cap B\rvert'$	#14	${}^{\lvert A\cap B\rvert'}/_{\lvert A\cup B\rvert'}$
#5	$\lvert A \cup B\rvert'$	#15	${}^{2\cdot\lvert A\cap B\rvert'}/_{\lvert A\rvert'+\lvert B\rvert'}$
#6	$\lvert A \setminus B\rvert'$	#16	${}^{\lvert A\cap B\rvert}/_{\min[\lvert A\rvert,\lvert B\rvert]}$
#7	$\lvert B \setminus A\rvert'$	#17	${}^{\lvert A\cap B\rvert'}/_{\max[\lvert A\rvert',\lvert B\rvert']}$
#8	$\lvert A \cup B - A \cap B\rvert'$	#18	${}^{\lvert A\cap B\rvert'}/_{\sqrt{\lvert A\rvert'\cdot\lvert B\rvert'}}$
#9	${}^{\lvert A-B\rvert'}/_{\lvert A\rvert'}$	#19	$\frac{\lvert A\cap B\rvert'+\lvert A\rvert'+\lvert B\rvert'}{2\cdot\lvert A\rvert'\cdot\lvert B\rvert'}$
#10	${}^{\lvert B-A\rvert'}/_{\lvert B\rvert'}$	#20	gold standard

Table 1: Feature set for regression

This approach can be used for replacing the **strm** coefficient by a similarity function learned from STS training data. The three features used in the previous experiment were extended to a total of 19 (see table 1) plus the gold standard as target. The feature #1 is the score of the corresponding text-similarity function described in the previous section. Three sets of features were constructed, each with 19 features using the soft cardinality in combination with the word-similarity functions **qgrams**, **csds** and **npmi**. Let's name these feature sets as *fs:qgrams*, *fs:csds* and *fs:npmi*. The submission labeled *run1* was obtained using the feature set *fs:qgrams* (19 features). The submission labeled *run2* was obtained using the aggregation of *fs:qgrams* and *fs:csds* (38 features). Finally, *run3* was the aggregation of *fs:grams*, *fs:csds* and *fs:npmi* (57 features).

7 Results in *SEM 2013 Shared Task

In this section three groups of systems are described by using the functions and models proposed in the previous sections. The first group (and simplest) of systems consist in using the scores of the three

Data set	$\mathbf{STS_{qgrams}}$	$\mathbf{STS_{csds}}$	$\mathbf{STS_{npmi}}$	*average*
headlines	**0.7625**	0.7243	0.7379	0.7562
OnWN	0.7022	0.7050	0.6832	**0.7063**
FNWM	0.2704	0.3713	**0.4215**	0.3940
SMT	0.3151	0.3325	**0.3408**	0.3402
mean	0.5570	0.5592	0.5653	**0.5747**
rank	8	7	6	**4**

Table 2: Unofficial results using text-similarity functions

Data set	*run1*	*run2*	*run3*
headlines	0.7591	0.7632	**0.7640**
OnWN	0.7159	0.7239	**0.7485**
FNWM	0.2806	**0.3679**	0.3487
SMT	0.2820	0.2786	**0.2952**
mean	0.5491	0.5586	**0.5690**
rank	14	8	**4**

Table 3: Unofficial results using linear regression

text-similarity functions $\mathbf{STS_{qgrams}}$, $\mathbf{STS_{csds}}$ and $\mathbf{STS_{npmi}}$. Table 2 shows the unofficial results of these three systems. The bottom row shows the positions that these systems would have obtained if they had been submitted to the *SEM shared task 2013. The last column shows the results of a system that combines the scores of three measures on a single score calculating the arithmetic mean. This is the best performing system obtained with the methods described in this paper.

Tables 2 and 4 show unofficial and official results of the method described in section 6 using linear regression and Bagging (30 iterations)+REP tree respectively. These results were obtained using WEKA (Hall et al., 2009).

Data set	*run1*	*run2*	*run3*
headlines	0.6410	**0.6713**	0.6603
OnWN	0.7360	**0.7412**	0.7401
FNWM	0.3442	**0.3838**	0.3347
SMT	**0.3035**	0.2981	0.2900
mean	0.5273	**0.5402**	0.5294
rank	23	**18**	22

Table 4: Official results of the submitted runs to STS *SEM 2013 shared task using Bagging + REP tree for regression

8 Discussion

Contrary to the observation we made in training data, the methods that used regression to predict the gold standard performed poorly compared with the text similarity functions proposed in Section 5. That is, the results in Table 2 overcome those in Tables 3 and 4. Also in training data, Bagging+REP tree surpassed linear regression, but, as can be seen in tables 3 and 4 the opposite happened in test data. This is a clear symptom of overfitting. However, the *OnWN* data set was an exception, which obtained the best results using linear regression. *OnWN* was the only one among the 2013 data sets that was not a surprise data set. Probably the 5.97% relative improvement obtained in *run3* by the linear regression versus the best result in Table 2 may be justified owing to some patterns discovered by the linear regressor in the OnWN'2012 training data which are projected on the OnWN'2013 test data.

Is worth noting that in all three sets of results, the lowest *mean* was consistently obtained by the text-overlapping methods, namely $\mathbf{STS_{qgrams}}$ and *run1*. The relative improvement in *mean* due to the use of distributional measures against the text-overlapping methods was 3.18%, 3.62% and 2.45% in each set of results (see Tables 2, 3 and 4). In *FNWM* data set, the biggest improvements achieved 55.88%, 31.11% and 11.50% respectively in the three groups of results, followed by *SMT* data set. Both in *FNWN* data set as in *SMT*, the texts are systematically longer than those found in *OnWN* and *headlines*. This result suggests that the improvement due to distributional measures is more significant in longer texts than in the shorter ones.

Lastly, it is also important to notice that the $\mathbf{STS_{qgrams}}$ text-similarity function obtained $mean = 0.5570$, which proved again to be a very strong text-overlapping baseline for the STS task.

9 Conclusions

We participated in the CORE-STS shared task in *SEM 2013 with satisfactory results obtaining positions 16, 20, and 21 in the official ranking. Our systems were based on a new parameterized resemblance coefficient derived from the Tversky's ratio model in combination with the soft cardinality. The three proposed text-similarity functions

used q-grams overlapping and distributional measures obtained from the ukWack corpus. These text-similarity functions would have been attained positions 5, 6 and 7 in the official ranking, besides a simple average of them would have reached the fourth place. Another important conclusion was that the plain text-overlapping method was consistently improved by the incremental use of the proposed distributional measures. This result was most noticeable in long texts.

In conclusion, the proposed text-similarity functions proved to be competitive despite their simplicity and the few resources used.

Acknowledgments

This research was funded by the Systems and Industrial Engineering Department, the Office of Student Welfare of the National University of Colombia, Bogotá, and through a grant from the Colombian Department for Science, Technology and Innovation Colciencias, proj. 110152128465. The Section 2 was proposed while the first author interned at Microsoft Research in 2012. The second author recognizes the support from Mexican Government (SNI, COFAA-IPN, SIP 20113295, CONACYT 50206-H) and CONACYT–DST India (proj. "Answer Validation through Textual Entailment").

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 19–27, Stroudsburg, PA, USA. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Gonzalez-Agirre Aitor. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval@*SEM 2012)*, Montreal,Canada. Association for Computational Linguistics.

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. Atlanta, Georgia, USA. Association for Computational Linguistics.

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 6th International Workshop on Semantic Evaluation (SemEval *SEM 2012)*, Montreal, Canada. 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.

Danushka Bollegala, Yutaka Matsuto, and Mitsuru Ishizuka. 2007. Measuring semantic similarity between words using web search engines. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 757–766, New York, NY, USA. ACM.

Leo Breiman. 1996. Bagging predictors. *Machine Learning*, 24(2):123–140.

Mark Hall, Frank Eibe, Geoffrey Holmes, and Bernhard Pfahringer. 2009. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1):10–18.

Sergio Jimenez and Alexander Gelbukh. 2012. Baselines for natural language processing tasks. *Appl. Comput. Math.*, 11(2):180–199.

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 *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval *SEM 2012)*, Montreal, Canada.

Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 60(5):493–502, October.

Michael D Lee, B.M. Pincombe, and Matthew Welsh. 2005. An empirical evaluation of models of text document similarity. *IN COGSCI2005*, pages 1254–1259.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *In AAAI'06*, pages 775–780.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet::Similarity: measuring the relatedness of concepts. In *Proceedings HLT-NAACL–Demonstration Papers*, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Phillip Resnik. 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633, October.

Frane Šaric, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašic. 2012. TakeLab: systems for measuring semantic text similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval *SEM 2012)*, Montreal, Canada. Association for Computational Linguistics.

Peter D. Turney. 2001. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In Luc De Raedt and Peter Flach, editors, *Machine Learning: ECML 2001*, number 2167 in Lecture Notes in Computer Science, pages 491–502. Springer Berlin Heidelberg, January.

Amos Tversky. 1977. Features of similarity. *Psychological Review*, 84(4):327–352, July.

I.H. Witten and E. Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisto, CA, 2nd edition.