UNAL-NLP: Combining Soft Cardinality Features for Semantic Textual Similarity, Relatedness and Entailment

Sergio Jimenez, George Dueñas,
and **Julia Baquero**
Universidad Nacional de Colombia
Ciudad Universitaria, edificio 453,
oficina 114, Bogotá, Colombia
[sgjimenezv,geduenasl,
jmbaquerov]@unal.edu.co

Alexander Gelbukh
Center for Computing Research (CIC),
Instituto Politécnico Nacional (IPN),
Av. Juan Dios Bátiz, Av. Mendizábal,
Col. Nueva Industrial Vallejo,
Mexico City, Mexico
www.gelbukh.com

Abstract

This paper describes our participation in the SemEval-2014 tasks 1, 3 and 10. We used an uniform approach for addressing all the tasks using the soft cardinality for extracting features from text pairs, and machine learning for predicting the gold standards. Our submitted systems ranked among the top systems in all the task and sub-tasks in which we participated. These results confirm the results obtained in previous SemEval campaigns suggesting that the soft cardinality is a simple and useful tool for addressing a wide range of natural language processing problems.

1 Introduction

The semantic textual similarity is a core problem in the computational linguistic field. Consequently, the previous evaluation campaigns of this task in SemEval have attracted the attention of many research groups worldwide (Agirre et al., 2012; Agirre et al., 2013).This year, 3 tasks related to this problem have been proposed exploring different facets such as semantic relatedness, entailment , multilingualism, lack of training data and imbalance in the amount of information.

The soft cardinality (Jimenez et al., 2010) is a simple concept that generalizes the classical set cardinality by considering the similarities among the elements in a collection for a more intuitive quantification of the number of elements in that collection. This approach can be applied to text applications representing texts as collections of words and providing a similarity function that compares two words. Varying this word-to-word similarity function the soft cardinality can reflect notions of syntactic similarity, semantic relatedness, among others. We (and others) have used this approach to address with success the semantic textual similarity and other tasks in previous SemEval editions (Jimenez et al., 2012b; Jimenez et al., 2012a; Jimenez et al., 2013a; Jimenez et al., 2013b; Jimenez et al., 2013c; Croce et al., 2013).

In this paper we describe our participating systems in the SemEval-2014 tasks 1, 3, and 10, which used the soft cardinality as core approach.

2 Features from Soft Cardinalities

The cardinality of a collection of elements is the counting of non-repeated elements in it. This definition is intrinsically associated with the notion of set, which is a collection of non-repeated elements.Thus, the cardinality of a collection or set A is denoted as $|A|$. Clearly, the cardinality of a collection with repeated elements treats groups of identical elements as a single instance contributing only with a unit (1) to the element counting. Jimenez et al. (2010) proposed the *soft cardinality* that uses a notion of similarity among elements for grouping not only identical elements but similar too. That notion of similarity among elements is provided by a similarity function that compares two elements a_i and a_j and returns a score in [0,1] interval, having $sim(a_i, a_i) = 1$. Although, it is not necessary that sim fulfills another metric properties aside of identity, symmetry is also desirable. Thus, the soft cardinality of a collection A, whose elements $a_1, a_2, \ldots, a_{|A|}$ are comparable with a similarity function $sim(a_i, a_j)$, is denoted as $|A|_{sim}$. This soft cardinality is given by the following expression:

$$|A|_{sim} = \sum_{i=1}^{|A|} \frac{w_{a_i}}{\sum_{j=1}^{|A|} sim(a_i, a_j)^p} \qquad (1)$$

It is trivial to see that $|A| = |A|_{sim}$ either if $p \to \infty$ or when the function sim is a crisp com-

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 732–742,
Dublin, Ireland, August 23-24, 2014.

Basic	Derived
$\|A\|$	$\|A \cap B\| = \|A\| + \|B\| - \|A \cup B\|$
$\|B\|$	$\|A \triangle B\| = \|A \cup B\| - \|A \cap B\|\|$
$\|A \cup B\|$	$\|A \setminus B\| = \|A\| - \|A \cap B\|$
	$\|B \setminus A\| = \|B\| - \|A \cap B\|$

Table 1: The 7 basic and derived cardinalities for two sets comparison.

parator, i.e. one that returns 1 for identical elements and 0 otherwise. This property shows that the soft cardinality generalizes the classical cardinality and that the parameter p controls its degree of "softness", whose default value is 1. The values w_{a_i} are optional "importance" weights associated with each element a_i, by default those weights can be assigned to 1.

For the tasks at hand, we represent each short text (lets say A) as a collection of words a_i and the sim function can be any operator that compares pairs of words. The motivation for using the soft cardinality is that the sim function can reflect any dimension of word similarity (e.g. syntactic, semantic) and the soft cardinality projects that notion at sentence level. For instance, if sim provides the degree of semantic relatedness between two words using WordNet, two texts A and B could be compared by computing $|A|_{sim}$, $|B|_{sim}$ and $|A \cup B|_{sim}$. Given that $A \cap B$ could be empty, the soft cardinality of the intersection must be approximated by $|A \cap B|_{sim} \approx |A|_{sim} + |B|_{sim} - |A \cup B|_{sim}$ instead of being computed directly from $A \cap B$ using equation 1. Using that approximation, the commonality (intersection) between A and B is induced by the pair-wise similarities provided by sim among the words in A and B.

Since more than a century when Jaccard (1901) proposed his well-known index, the classical set cardinality has been used to build similarity functions for set comparison. Any binary-cardinality-based similarity function is an algebraic combination of $|A|$, $|B|$ and either $|A \cap B|$ or $|A \cup B|$ (e.g. Jaccard, Dice, Tversky, overlap and cosine indexes). These three cardinalities describes unambiguously all the regions in the Venn's diagram when comparing two sets. Thus, in this scenario 4 possible cardinalities can be derived from these 3 basic cardinalities, see Table 1. Clearly, the same set of cardinalities can be obtained for the soft cardinality.

When training data is available, which is the case for tasks 1, 3 and 10 in SemEval 2014, it is possible to think that instead of using an ad-hoc expression (e.g. Jaccard, Dice) the similarity function can be obtained using the cardinalities in Table 1 as features for a machine-learning regression algorithm. Our hypothesis is that such learnt function should predict in a more accurate way the gold standard variable than any other ad-hoc function. However, these cardinality features are intrinsically correlated with the length of the texts where they were obtained. This correlation makes that the performance of the learnt similarity function could be dependent of the length of the texts. For instance, if the function was trained using long texts it is plausible to think that this function would be more effective when tested with long texts than with shorter ones. Having this in mind, an extended set of rational features is proposed, whose values are standardized in [0,1] interval aiming to reduce the effect of the length of the texts. These features are presented in Table 2.

#	Feature expression
1	$\|A\| / \|A \cup B\|$
2	$\|A\| - \|A \cap B\| / \|A\|$
3	$\|A\| - \|A \cap B\| / \|A \cup B\|$
4	$\|B\| / \|A \cup B\|$
5	$\|B\| - \|A \cap B\| / \|B\|$
6	$\|B\| - \|A \cap B\| / \|A \cup B\|$
7	$\|A \cap B\| / \|A\|$
8	$\|A \cap B\| / \|B\|$
9	$\|A \cap B\| / \|A \cup B\|$
10	$\|A \cup B\| - \|A \cap B\| / \|A \cap B\|$

Table 2: Extended set of 10 rational features.

The soft cardinality has proven to overcome the classic cardinality in the semantic textual similarity (STS) task in previous SemEval campaigns (Jimenez et al., 2012b; Jimenez et al., 2013a). Even using a simplistic function sim based on q-grams of characters, the soft cardinality method ranked third among 89 participating systems (Agirre et al., 2012). Thus, our participating systems in the SemEval 2014 campaign were based on the previously described set of 17 features, obtained from the soft cardinality with different sim functions for comparing pairs of words. Each sim function produced a different set of features, which were combined with a regression algorithm for similarity and relatedness tasks. Similarly, a classification algorithm was used for the

entailment task.

3 Systems Description

In this section the different feature sets used for each submitted system to the different task and subtask are described. Besides, the data used for training, parameters and other preprocessing details are described for each system.

3.1 Task 1: Textual Relatedness and Entailment

The task 1 is based on the SICK (Sentences Involving Compositional Knowledge) data set (Marelli et al., 2014), which contains nearly 10,000 pairs of sentences manually labeled by relatedness and entailment. The relatedness gold labels range from 1 to 5, having 1 the minimum level of relatedness between the texts and 5 for the maximum. The entailment labels have three categorical values: *neutral*, *contradiction* and *entailment*. The two sub tasks consist of predicting the relatedness and entailment gold standards using approximately the 50% of the text pairs as training and the other part as test bed.

Our overall approach consists in extracting 4 different sets of features using the method presented in section 2 and training a machine learning algorithm for predicting the gold standard labels in the test data. Each feature set is described in the following 4 subsections and the subsection 3.1.6 provides details of the used combination of features, machine learning algorithm and preprocessing details.

3.1.1 String-Matching Features

First, all texts in the SICK data set where preprocessed by lower casing, tokenizing and stop-word removal (using the NLTK[1]). Then each word was reduced to its stem using the Porter's algorithm (Porter, 1980) and a *idf* weight (Jones, 2004) was associated to each stem (w_{a_i} weights in eq. 1) using the very SICK data set as document collection. Next, for each instance in the data, which is composed of two texts A and B, the 17 features listed in Tables 1 and 2 where extracted using eq.1. The used word-to-word similarity function sim decomposes each word in bags of 3-grams of characters, which are compared using the symmetrical Tversky's index (Tversky, 1977; Jimenez et al., 2013a). Thus, the similarity between two pairs of words w_1 and w_2, represented each one as a collection of 3-grams of characters, is given by the following expression:

$$sim(w_1, w_2) = \frac{|c|}{\beta(\alpha|w_{min}| + (1-\alpha)|w_{max}|) + |c|} \quad (2)$$

$$|c| = |w_1 \cap w_2| + bias_{sim},$$
$$|w_{min}| = \min[|w_1 \setminus w_2|, |w_2 \setminus w_1|],$$
$$|w_{max}| = \max[|w_1 \setminus w_2|, |w_2 \setminus w_1|].$$

The values used for the parameters were $\alpha = 1.9$, $\beta = 2.36$, $bias = -0.97$, and $p = 0.39$ (where *p* corresponds to eq.1). The motivation and justification for these parameters can be found in (Jimenez et al., 2013a). These values were obtained by building a text similarity function using the Dice's coefficient and the soft cardinalities plugging eq.2 in eq.1. Next, this text similarity function is evaluated in the 5,000 training text pairs and the obtained scores are compared against the relatedness gold-standard using the Pearson's correlation.

w_{a_i} are not training parameters, but they are weights associated with the words. These weights could have been obtained from a larger corpus, but we use the training texts to obtain them. This process is repeated iteratively exploring the search space defined by these 4 parameters using a hill-climbing approach until a maximum correlation is reached. We observe that the optimal values of the parameters p, α, β, and $bias$ vary considerably between the data sets and for the different *sim* functions of word-to-word similarity. We do not yet understand from which factors of the data and the *sim* functions depend on these parameters. This issue will be the objective of further research.

Henceforth, the set of 17 string-based features described in this subsection will be referred as **SM**.

3.1.2 ESA Features

For this set of features we used the idea proposed by Gabrilovich and Markovitch (2007) of enriching the representation of a text by representing each word by its textual definition in a knowledge base, i.e. explicit semantic analysis (ESA). For that, we used as knowledge base the synset's textual definitions provided by WordNet. First, in order to determine the textual definition associated to each word, the texts were tagged using

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

the maximum entropy POS tagger included in the NLTK. Next, the adapted Lesk algorithm (Banerjee and Pedersen, 2002) for word sense disambiguation was applied in the texts disambiguating one word at the time. The software package used for this disambiguation process was *pywsd*[2]. The arguments needed for the disambiguation of each word are the POS tag of the target word and the entire sentence as context. Once all the words are disambiguated with their corresponding WordNet synsets, each word is replaced by all the words in their textual definition jointly with the same word and its lemma. The final result of this stage is that each text in the data set is replaced by a longer text including the original text and some related words. The motivation of this procedure is that the extended versions of each pair of texts have more chance of sharing common words that the original texts.

The extended versions of these texts were used to obtain another 17 features with the same procedure described in the previous subsection (3.1.1). This feature subset will henceforth be referred as **ESA**.

3.1.3 Features for each part-of-speech category

This set of features is motivated by the idea proposed by Corley and Mihalcea (2005) of grouping words by their POS category before being compared for semantic textual similarity. Our approach consist in provide a version of each text pair in the data set for each POS category including only the words belonging to that category. For instance, the pair of texts {"*A beautiful girl is playing tennis*", "*A nice and handsome boy is playing football*"} produce new pairs such as: {"*beautiful*", "*nice handsome*"} for the ADJ tag, {"*girl tennis*", "*boy football*"} for NOUN and {"*is playing*", "*is playing*"} for VERB.

Again, the POS tags were provided by the NLTK's max entropy tagger. The 28 POS categories were simplified to 9 categories in order to avoid an excessive number of features and hence sparseness; the used mapping is shown in Table 3. Next, for each one of the 9 new POS categories a set of 17 features (**SM**) is extracted reusing again the method proposed in subsection 3.1.1. The only difference with the method described in that subsection is that the stop-words were not removed and the stemming process was not performed. The motivation for generating this feature sets by POS category is that the machine learning algorithms could weight differently each category. The intuition behind this is that it is reasonable that categories such as VERB and NOUN could play a more important role for the task at hand than others such as ADJ or PREP. Using these categorized features, such discrimination among POS categories can be discovered from the training data.

Reduced tag set	NLTK's POS tag set
ADJ	JJ,JJR,JJS
NOUN	NN,NNP,NNPS,NNS
ADV	RB,RBR,RBS,WRB
VERB	VB,VBD,VBG,VBN,VBP,VBZ
PRO	WP,WP$,PRP,PRP$
PREP	RP,IN
DET	PDT,DT,WDT
EX	EX
CC	CC

Table 3: Mapping reduction of the POS tag set.

Finally, the total number of features in this set is 153 (17 features $\times$ 9 POS categories). This feature set will be referred as **POS**.

3.1.4 Features From Dependencies

The *syntactic soft cardinality* (Croce et al., 2012; Croce et al., 2013) extend the soft cardinality approach by representing texts as bags of dependencies instead of bags of words. Each dependency is a 3-tuple composed of two syntactically related words and the type of their relationship. For instance, the sentence "*The boy plays football*" can be represented with 3 dependencies: [**det**,"*boy*","*The*"], [**subj**,"*plays*","*boy*"] and [**obj**,"*plays*","*football*"]. Clearly, this representation distinguish pairs of texts such as {"*The dog bites a boy*","*The boy bites a* dog"}, which are indistinguishable when they are represented as bags of words. This representation can be obtained automatically using the Stanford Parser (De Marneffe et al., 2006), which in addition provides a dependency identifying the root word in a sentence. We used the version 3.3.1[3] of that parser to obtain such representation.

Once the texts are represented as bags of dependencies, it is necessary to provide a similarity function between two dependency tuples in or-

[2]https://github.com/alvations/pywsd

[3]http://nlp.stanford.edu/software/lex-parser.shtml

der to use the soft cardinality (eq. 1) and hence to obtain the 17 cardinality features in Tables 1 and 2. Such function can be obtained using the sim function (eq. 2) for comparing the first and second words between the dependencies and even the labels of the dependency types. Let's consider two dependencies tuples $d = [d_{dep}, d_{w_1}, d_{w_2}]$ and $p = [p_{dep}, p_{w_1}, p_{w_2}]$ where d_{dep} and p_{dep} are the labels of the dependency type; d_{w_1} and p_{w_1} are the first words on each dependency tuple; and d_{w_2} and p_{w_2} are the second words. The similarity function for comparing two dependency tuples can be a linear combination of the sim scores between the corresponding elements of the dependency tuples by the following expression:

$$sim_{dep}(d, p) = \gamma sim(d_{dep}, p_{dep}) + \delta sim(d_{w_1}, p_{w_2}) + \lambda sim(d_{w_2}, p_{w_2})$$

Although, it is unusual to compare the dependencies' type labels d_{dep} and p_{dep} with a similarity function designed for words, we observed experimentally that this approach yield better overall performance in the relatedness task in comparison with a simple crisp comparison. The optimal values for the parameters $\gamma = -3$, $\delta = 10$ and $\lambda = 3$ were determined with the same methodology used in subsection 3.1.1 for determining α, β and $bias$. Clearly, the fact that $\delta > \lambda$ means that the first words in the dependency tuples plays a more important role than the second ones for the task at hand. However, the fact that $\gamma < 0$ is counter intuitive because it means that the lower the similarity between the dependency type labels is, the larger the similarity between the two dependencies. Up to date we have been unable to find a plausible explanation for this phenomenon. This set of 17 features will be referred hereinafter as **DEP**.

3.1.5 Additional Features

In addition to the feature sets based in soft cardinality, we designed some features aimed to address linguistic phenomena such as antonymy, hypernymy and negation.

Antonymy: Consider the following text pair from the test data {"*A man is emptying a container made of plastic*","*A man is filling a container made of plastic*" }, which is labeled as a *contradiction* with a relatedness score of 3.91. Clearly, these labels are explained by the antonymy relation between "*emptying*" and "*filling*". Given that none of the features presented above address this issue, a list of 11,028 pairs of antonym words was gathered from several web sites (see Table 4) and from the antonymy relationships in WordNet, in order to detect these cases. That list was used to count the number of occurrences of pairs antonym words between pairs of texts and in each one of the texts. Thus, for any pair of texts A and B (represented as sets of words), three features (referred henceforth as **ANT**) were extracted:

antonym_AB Counts the number of occurrences of pairs of antonyms in $A \times B$ (Cartesian product) or in $B \times A$.

antonym_AA Counts the number of occurrences of pairs of antonyms in $A \times A$.

antonym_BB Counts the number of occurrences of pairs of antonyms in $B \times B$.

Hypernymy: Consider the following text pair from the test data {"*A man is sitting comfortably at a table*","*A person is sitting comfortably at the table*" }, which is labeled as an *entailment* with a relatedness score of 3.96. In this case, the entailment is based on the hypernymy between "*person*" and "*man*". In order to capture this linguistic factor 3 features similar to the previously described antonym features were proposed. First, word sense disambiguation was performed (as described in subsection 3.1.2) for obtaining a synset label for each word. Secondly, we build a binary function $hyp(ss_1, ss_2)$ that takes two WordNet synsets as arguments and returns 1 if ss_1 is a hypernym of ss_2 with a maximum depth in the WordNet's is-a hierarchy of 6 steps, and 0 otherwise. This hypernymy function was build using the WordNet interface provided by the NLTK. Next, based on that synset-to-synset function, a text-to-text function that captures the degree or hypernymy in a text or in a pair of texts was build using the Monge-Elkan measure (Monge and Elkan, 1996). Thus, for two texts A and B represented as sets of synset labels, the following expression measures their degree of hypernymy:

$$HYP(A, B) = \frac{1}{|A|} \sum_{i=1}^{|A|} \max_{j=1}^{|B|} hyp(a_i, b_j)$$

Using the function $HYP(*, *)$, 3 features are extracted from each pair of text (referred henceforth as **HYP**):

hypernym_AB from $HYP(A, B)$

http://www.myenglishpages.com/site_php_files/vocabulary-lesson-opposites-adjectives.php
http://www.allaboutspace.com/wordlist/opposites.shtml
http://www.michigan-proficiency-exams.com/antonym-list.html
http://examples.yourdictionary.com/examples-of-antonyms.html
http://www.synonyms-antonyms.com/antonyms.html
http://englishwilleasy.com/word-must-know/vocabulary/vocabulary-list-by-opposites-or-antonyms/
http://www.meridianschools.org/staff/districtcurriculum/moreresources/languagearts/all grades/antonyms.doc
http://mrsbrower.weebly.com/uploads/1/3/2/4/13243672/antonymlist.pdf
https://foxhugh.wordpress.com/word-lists/list-of-antonyms/
http://www.paulnoll.com/Books/Clear-English/English-antonyms-1.html
http://wordnet.princeton.edu/wordnet/download/

Table 4: URLs used for the list of 11,028 antonym pairs (accessed on March 20, 2014).

hypernym_AA from $HYP(A, A)$

hypernym_BB from $HYP(B, B)$

Negation: Negations play an important role in the task at hand. For instance, consider this pair of texts {"*A person is rinsing a steak with water*","*A man is not rinsing a large steak*"} labeled as a *contradiction*. In that example the negation of the verb "*rising*" is the main factor of contradiction. In order to capture this linguistic feature we build a simple function that detects the occurrence of a verb negation if the text contains one of the following words: "*not*", "*n't*", "*nor*", "*null*", "*neither*", "*either*", "*barely*", "*scarcely*" and "*hardly*". Similarly, noun negation is detected looking for the words: "*no*", "*none*", "*nobody*", "*nowhere*", "*nothing*" and "*never*". Thus, for two texts A and B, 4 features are extracted (referred henceforth as **NEG**):

verb_neg_A if verb negation is detected in A

verb_neg_B if verb negation is detected in B

noun_neg_A if noun negation is detected in A

noun_neg_B if noun negation is detected in B

3.1.6 Submitted Runs and Results

RUN1 (PRIMARY) This system produced predictions by extracting all the features described previously (**SM**, **ESA**, **POS**, **DEP**, **ANT**, **HYP** and **NEG**) from all the texts in the SICK data set. Next, two machine learning models were obtained (WEKA (Hall et al., 2009) was used for that) using the training part of SICK, one for regression (relatedness) and another for classification (entailment). The regression model was a *reduced-error pruning tree* (*REPtree*) (Quinlan, 1987) boosted with 20 iterations of *bagging (Breiman, 1996)*. The classification model was a *J48Graft* tree also boosted with 20 bagging iterations. These two models produced the predictions for the test part of SICK.

RUN2 This system is similar to the one used in RUN1, but it used only the feature sets **SM** and **NEG**. Another difference is that a linear regression was used instead of the *REPtree* and no *bagging* was performed.

RUN3 The same as RUN1, but again, linear regression was used instead of the *REPtree* and no *bagging* was performed.

RUN4 The same as RUN2, but the models were boosted with 20 iterations of *bagging*.

RUN5 The same as RUN3, but 30 iterations of bagging were used instead of 20.

The official results obtained by these systems (prefixed UNAL-NLP) are shown in Table 5 jointly with those obtained by other 3 top systems among the 18 participating systems. Our primary run (RUN1) obtained pretty competitive results ranking 3th and 4th in the entailment and relatedness tasks. The RUN4 obtained a remarkable performance (it would be ranked 6th for entailment and 8th for relatedness) in spite of the fact that is a system purely based on string matching. The comparison of our runs 1, 3 and 5, which mainly differs by the use of *bagging*, shows that this boosting method provides considerable improvements. In fact, comparing RUN3 (all features, no bagging) and RUN4 (SM and NEG feature sets boosted with bagging), they performed similarly in spite of the considerable larger number of features used in RUN3. Besides, the RUN5 slightly outperformed our primary run (RUN1) us-

	Entailment		Relatedness			
system	**accuracy**	**official rank**	**Pearson**	**Spearman**	**MSE**	**official rank**
UNAL-NLP_run1 (primary)	83.05%	3rd/18	0.8043	0.7458	0.3593	4th/17
UNAL-NLP_run2	79.81%	-	0.7482	0.7033	0.4487	-
UNAL-NLP_run3	80.15%	-	0.7747	0.7286	0.4081	-
UNAL-NLP_run4	80.21%	-	0.7662	0.7142	0.4210	-
UNAL-NLP_run5	83.24%	-	0.8070	0.7489	0.3550	-
ECNU_run1	83.64%	2nd/18	**0.8280**	**0.7689**	0.3250	1st/17
Stanford_run5	74.49%	12th/18	0.8272	0.7559	**0.3230**	2nd/17
Illinois-LH_run1	**84.58%**	1st/18	0.7993	0.7538	0.3692	5th/17

Table 5: Results for task 1.

ing 10 additional iterations of bagging.

3.1.7 Error Analysis

Our primary run for the task 1 failed in 835 pairs of sentences out of 4,927 in the entailment subtask. We wanted to understand in why our system failed in these 835 instances, so we classified manually these instances in 4 error categories (each instance could be assigned to several categories).

Paraphrase not detected (NP): example={"*Two groups of people are playing football*", "*Two teams are competing in a football match*"}, gold standard=*entailment*, prediction=*neutral*, number of occurrences= 420 (50.3%). The system failed to detect the paraphrase between "*groups of people*" and "*teams*".

Negation not detected (NN) : example={"*There is no one playing the guitar*", "*Someone is playing the guitar*"}, gold standard=*contradiction*, prediction=*neutral*, number of occurrences=94 (11.3%). The system failed to detect that the contradiction is due to the negation in the first text.

False similarity between words (NSS) : example={"*Two dogs are playing by a tree*", "*Two dogs are sleeping by a tree*"}, gold standard=*neutral*, prediction=*entailment*, number of occurrences=413 (49.5%). The only difference between these 2 sentences is the gerund "*playing*" vs. "*sleeping*", which the system erroneously considered as similar.

Antonym not detected (NA): example={"*Three children are running down hill*", "*Three children are running up hill*"}, gold standard=*contradiction*, prediction=*entailment*, number of occurrences=40 (4.8%). The only difference between these 2 sentences is the words "*down*" vs. "*up*". In spite that this pair of antonyms was included in the antonym list, the system failed to distinguish the contradiction between the texts.

Error category	**NP**	**NN**	**NSS**	**NA**
NP	420	5	125	0
NN	-	94	1	0
NSS	-	-	413	22
NA	-	-	-	40

Table 6: Co-ocurrences of types of errors in RUN1 (task1).

The matrix in Table 6 reports the number of co-occurrences of error categories in the 835 instances erroneously classified.

3.2 Task 3: Cross-level Semantic Similarity

The SemEval 2014 task 3 (cross-level semantic similarity) (Jurgens et al., 2014) proposed the semantic textual similarity task but across different textual levels, namely *paragraph-to-sentence*, *sentence-to-phrase*, *phrase-to-word* and *word-to-sense*. As usual, the goal is to predict the gold similarity scores for each pair of texts. For each one of these cross-level comparison types there were proposed a separated training and test data sets. Basically, we addressed this task using the set of features **SM** presented in subsection 3.1.1 in combination with a text expansion approach similar to the method presented in subsection 3.1.2.

3.2.1 Paragraph-to-sentence and Sentence-to-phrase

For these two cross-level comparison types we extracted the **SM** feature set using the provided texts. The model parameters obtained for paragraph-to-sentence were $\alpha = 0.1$, $\beta = 1.75$, $bias = -1.35$, $p = 1.55$; and for sentence-to-phrase were $\alpha = 0.68$, $\beta = 0.92$, $bias = -0.92$, $p = 2.49$.

The system for the RUN2 used the **SM** feature set and a machine learning model build with the provided training data for generating the similarity score predictions for the test data. For the paragraph-to-sentence data set the model was a *REPtree* for regression boosted with 40 *bagging* iterations. Similarly, the model for the sentence-to-phrase data set was a linear regressor also boosted with 40 *bagging* iterations.

Unlike RUN2, RUN1 does not make use of any machine learning algorithm. Instead, we used the only the basic cardinalities (see Table 1) from the **SM** feature set in combination with an ad-hoc resemblance coefficient, i.e. the Dice's coefficient $2|A \cap B| / |A| + |B|$ for the paragraph-to-sentence data set. In turn, for sentence-to-phrase the overlap coefficient, i.e. $|A \cap B| / \min[|A|, |B|]$, was used.

3.2.2 Phrase-to-word and Word-to-sense

Before applying the same procedure used in the previous subsection, the texts in the phrase-to-word and word-to-sense data sets were expanded with a similar approach to that was used in subsection 3.1.2.

Phrase-to-word expansion: First, the "word" was expanded finding its corresponding WordNet synset using the adapted Lesk's algorithm providing as context the "phrase". Then, once the word's synset is obtained, the "word" text is extended with the textual definition of the synset. Similarity, this procedure is repeated for each word in the "phase" obtaining and extended version of the phrase. Finally, these two texts are used for extracting the **SM** feature set. The model parameters were $\alpha = 0.8$, $\beta = 1.9$, $bias = -0.8$, $p = 1.5$.

Word-to-sense expansion: First, the "sense" (i.e. synset) is replaced by its textual definition and its lemma. At this point the pair word-sense becomes a pair word-sentence. Then, the synset of the "word" is obtained performing the adapted Lesk's algorithm. Next, the "word" is extended with textual definition of the synset. Finally, these two texts are used for extracting the **SM** feature set obtaining the following model parameters were $\alpha = 0.59$, $\beta = 0.9$, $bias = -0.89$, $p = 3.91$.

3.2.3 Results

The official results obtained by the two submitted runs jointly with other 3 top systems are shown in Table 7. Our submissions (prefixed with UNAL-NLP) ranked 3rd and 5th among 38 participating systems, showing that the **SM** (string-matching) feature set is effective for the prediction of similarity scores. Particularly, in the paragraph-to-sentence data set, which has the longest text, RUN2 obtained the best official score. In contrast, the scores obtained for the phrase-to-word and word-to-sense data sets were considerably lower in comparison with the top system, but still competitive against most of the other participating systems.

test data	train data
OnWN (en)	OnWN 2012/2013 test
headlines (en)	headlines 2013 test
images (en)	MSRvid 2012 train and test
deft-news (en)	MSRpar 2013 train and test
deft-forum (en)	MSRvid 2012 train and test OnWN 2012/2013 test
tweet-news (en)	SMTeuroparl 2012 test SMTnews 2012 test
Wikipedia (es)	SMTeuroparl 2012 train
news (es)	SMTeuroparl 2012 train

Table 8: Training data used for the STS-2014 data sets (task 10).

3.3 Task 10: Multilingual Semantic Similarity

The SemEval-2014 task 10 (multilingual semantic similarity) (Agirre et al., 2014) is the sequel of the semantic textual similarity (STS) evaluations at SemEval in the past two years (Agirre et al., 2012; Agirre et al., 2013). This year 6 test data sets were proposed in English and 2 data sets in Spanish. Similarly to the 2013 campaign, there is not explicit training data for each data set. Consequently, different data sets from the previous STS evaluations were selected to be used as training data for the new data sets. The selection criterion was the average character length and type of the texts. The Table 8 shows the training data used for each test data set.

3.3.1 English Subtask

The RUN1 for the English data sets was produced with a parameterized similarity function based on the **SM** feature set and the symmetrized Tversky's index (Tversky, 1977; Jimenez et al., 2013a). For a detailed description of this function and its parameters, please refer to the $\mathbf{STS_{sim}}$ feature in the system description paper of the NTNU team (Lynum et al., 2014). The parameters used in that

System	Para-2-Sent	Sent-2-Phr	Phr-2-Word	Word-2-Sense	Official Rank
SimCompass_run1	0.811	0.742	**0.415**	**0.356**	1st/38
ECNU_run1	0.834	**0.771**	0.315	0.269	2nd/38
UNAL-NLP_run2	**0.837**	0.738	0.274	0.256	3rd/38
SemantiKLUE_run1	0.817	0.754	0.215	0.314	4th/38
UNAL-NLP_run1	0.817	0.739	0.252	0.249	5th/38

Table 7: Official results for task 3 (Pearson's correlation).

Data	α	β	$bias$	p	α'	β'	$bias'$
OnWN	0.53	-0.53	1.01	1.00	-4.89	0.52	0.46
headlines	0.36	-0.29	4.17	0.85	-4.50	0.43	0.19
images	1.12	-1.11	0.93	0.64	-0.98	0.50	0.11
deft-news	3.36	-0.64	1.37	0.44	2.36	0.72	0.02
deft-forum	1.01	-1.01	0.24	0.93	-2.71	0.42	1.63
tweet-news	0.13	0.14	2.80	0.01	2.66	1.74	0.45

Table 9: Optimal parameters used for task 10 in English.

function are reported in Table 9. Unlike subsection 3.1.1 where the Dice's coefficient was used as the text similarity function, here the symmetrical Tversky's index (eq. 2) was reused generating the three additional parameters marked with apostrophe (α', β' and $bias'$).

For the RUN2 the $\mathbf{SM}$ feature set was extracted from all the data sets in English (en) listed in Table 8. Then, a *REPtree* (Quinlan, 1987) boosted with 50 *bagging* iterations (Breiman, 1996) was trained using the training data sets selected for each test data set. Finally, these machine learning models produced the similarity score predictions for each test data set.

The RUN3 was identical to the RUN2 but included additional feature sets apart from $\mathbf{SM}$, namely: $\mathbf{ESA}$, $\mathbf{POS}$ and $\mathbf{WN}$. The $\mathbf{WN}$ feature set is the same as $\mathbf{SM}$, but replacing the word-to-word similarity function in eq. 2 by the *path* measure from the WordNet::Similarity package (Pedersen et al., 2004).

3.3.2 Spanish Subtask

The Spanish system was based entirely in the $\mathbf{SM}$ feature set with some small changes for adapting the system to Spanish. Basically, the list of English stop-words was replaced by the Spanish stop-words provided by the NLTK. In addition, the Porter stemmer was replaced by its Spanish equivalent, i.e. the Snowball stemmer for Spanish. The RUN1 is equivalent to the RUN1 for the English subtask described in the previous subsection. The parameters used for the text similarity function were $\alpha = 1.16$, $\beta = 1.08$, $bias = 0.02$, $p = 1.02$, $\alpha' = 1.54$, $\beta' = 0.08$ and $bias' = 1.37$. The description and meaning of these parameters can be found in (Lynum et al., 2014) associated to the $\mathbf{STS_{sim}}$ feature.

data set	run1	run2	run3
deft-forum	0.5043	0.3826	0.4607
deft-news	0.7205	0.7305	0.7216
headlines	0.7616	0.7645	0.7605
images	0.8071	0.7706	0.7782
OnWN	0.7823	0.8268	0.8426
tweet-news	0.6145	0.4028	0.6583
mean (en)	0.7113	0.6573	0.7209
official rank (en)	12th/38	22th/38	9th/38
Wikipedia	**0.7804**	0.7566	0.6894
news	0.8154	0.7829	0.7965
mean (es)	0.8013	0.7723	0.7533
official rank (es)	3rd/22	9th/22	12th/22

Table 10: Official results for the task 10 (Pearson's correlation).

The RUN2 was obtained using the $\mathbf{SM}$ feature set and a linear regressor for generating the similarity score predictions. Similarity, RUN3 used the same feature set SM in combination with a *REPtree* boosted with 30 *bagging* iterations.

3.3.3 Results

The results for the 3 submitted runs corresponding to the 2 sub tasks (English and Spanish) are shown in Table 10. It is important to note that the RUN1 for the Wikipedia data set in Spanish was the top system among 22 participating systems. This result is remarkable given that this system was trained with a data set in English showing the domain adaptation ability of the soft cardinality approach.

4 Conclusions

We participated in the SemEval-2014 task 1, 3 and 10 with an uniform approach based on soft cardinality features, obtaining pretty satisfactory results in all data sets, tasks and sub tasks. This approach has been used since SemEval-2012 in all versions of the following tasks: semantic textual similarity (Jimenez et al., 2012b; Jimenez et al., 2013a), typed similarity (Croce et al., 2013), cross-lingual textual entailment (Jimenez et al., 2012a; Jimenez et al., 2013c), student response analysis (Jimenez et al., 2013b), and multilingual semantic textual similarity (Lynum et al., 2014). In the majority of the cases, the systems based on soft cardinality, built by us and other teams, have been among the top systems. Given the uniformity of the approach, the consistency of the results, the few computational resources required and the overall conceptual simplicity, the soft cardinality is established as a useful tool for a wide spectrum of applications in natural language processing.

References

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.

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.

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

Satanjeev Banerjee and Ted Pedersen. 2002. An adapted lesk algorithm for word sense disambiguation using WordNet. In *Computational linguistics and intelligent text processing*, page 136–145. Springer.

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

Courtney Corley and Rada Mihalcea. 2005. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, EMSEE '05, page 13–18, Stroudsburg, PA, USA.

Danilo Croce, Valerio Storch, P. Annesi, and Roberto Basili. 2012. Distributional compositional semantics and text similarity. In *2012 IEEE Sixth International Conference on Semantic Computing (ICSC)*, pages 242–249, September.

Danilo Croce, Valerio Storch, and Roberto Basili. 2013. UNITOR-CORE TYPED: Combining text similarity and semantic filters through SV regression. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: SemanticTextual Similarity*, page 59, Atlanta, Georgia, USA.

Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, page 449–454, Genoa, Italy, May.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, page 1606–1611, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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

Paul Jaccard. 1901. Etude comparative de la distribution florare dans une portion des alpes et des jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, pages 547–579.

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. 2012a. Soft cardinality: A parameterized similarity function for text comparison. In *SemEval 2012*, Montreal, Canada.

Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. 2012b. Soft cardinality+ ML: Learning adaptive similarity functions for cross-lingual textual entailment. In *SemEval 2012*, Montreal, Canada.

Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. 2013a. SOFTCARDINALITY-CORE: Improving text overlap with distributional measures for semantic textual similarity. In **SEM 2013*, Atlanta, Georgia, USA, June.

Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. 2013b. SOFTCARDINALITY: Hierarchical text overlap for student response analysis. In *SemEval 2013*, Atlanta, Georgia, USA, June.

Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. 2013c. SOFTCARDINALITY: Learning to identify directional cross-lingual entailment from cardinalities and SMT. In *SemEval 2013*, Atlanta, Georgia, USA, June.

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

David Jurgens, Mohammad T. Pilehvar, and Roberto Navigli. 2014. SemEval-2014 task 3: Cross-level semantic similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, Dublin, Ireland, August.

André Lynum, Partha Pakray, Björn Gambäck, and Sergio Jimenez. 2014. NTNU: Measuring semantic similarity with sublexical feature representations and soft cardinalty. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, Dublin, Ireland, August.

Marco Marelli, Stefano Menini, Marco Baroni, Lucia Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of LREC*, Reykjavik, Iceland, May.

Alvaro E. Monge and Charles Elkan. 1996. The field matching problem: Algorithms and applications. In *Proceeding of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 267–270, Portland, OR.

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

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

J. Ross Quinlan. 1987. Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234.

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