NTNU: Measuring Semantic Similarity with Sublexical Feature Representations and Soft Cardinality

André Lynum, Partha Pakray, Björn Gambäck
{andrely,parthap,gamback}@idi.ntnu.no
Norwegian University of Science and Technology
Trondheim, Norway

Sergio Jimenez
sgjimenezv@unal.edu.co
Universidad Nacional de Colombia
Bogotá, Colombia

Abstract

The paper describes the approaches taken by the NTNU team to the SemEval 2014 Semantic Textual Similarity shared task. The solutions combine measures based on lexical soft cardinality and character n-gram feature representations with lexical distance metrics from TakeLab's baseline system. The final NTNU system is based on bagged support vector machine regression over the datasets from previous shared tasks and shows highly competitive performance, being the best system on three of the datasets and third best overall (on weighted mean over all six datasets).

1 Introduction

The Semantic Textual Similarity (STS) shared task aims at providing a unified framework for evaluating textual semantic similarity, ranging from exact semantic equivalence to completely unrelated texts. This is represented by the prediction of a similarity score between two sentences, drawn from a particular category of text, which ranges from 0 (different topics) to 5 (exactly equivalent) through six grades of semantic similarity (Agirre et al., 2013). This paper describes the NTNU submission to the SemEval 2014 STS shared task (Task 10). The approach is based on the lexical and distributional features of the baseline TakeLab system from the 2012 shared task (Šarić et al., 2012), but improves on it in three ways: by adding two new categories of features and by using a bagging regression model to predict similarity scores.

The new feature categories added are based on soft cardinality and character n-grams, described in Section 2. The parameters of the two categories are optimised over several corpora and the features are combined through support vector regression (Section 3) to create the actual systems (Section 4). As Section 5 shows, the new measures give the baseline system a substantial boost, leading to very competitive results in the shared task evaluation.

2 Feature Generation Methods

The methods used for creating new features utilise soft cardinality and character n-grams. Soft cardinality (Jimenez et al., 2010) was used successfully for the STS task in previous SemEval editions (Jimenez et al., 2012a; Jimenez et al., 2013a). The NTNU systems utilise an ensemble of such 18 measures, based only on surface text information, which were extracted using soft cardinality with different similarity functions, as further described in Section 2.1.

Section 2.2 then introduces the similarity measures based on character n-gram feature representations, which proved themselves as the strongest features in the STS 2013 task (Marsi et al., 2013). The measures used here replace character n-gram features with cluster frequencies or vector values based on the n-gram collocational structure learned in an unsupervised manner from text data. A variety of n-gram feature representations were trained on subsets of Wikipedia and the best performing ones were used for the new measures, which are based on cosine similarity between the document vectors derived from each sentence in a given pair.

2.1 Soft Cardinality Measures

Soft cardinality resembles classical set cardinality as it is a method for counting the number of elements in a set, but differs from it in that similarities among elements are being considered for the "soft counting". The soft cardinality of a set of words

$A = \{a_1, a_2, .., a_{|A|}\}$ (a sentence) is defined by:

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

Where p is a parameter that controls the cardinality's softness (p's default value is 1) and w_{a_i} are weights for each word, obtained through inverse document frequency (*idf*) weighting. $sim(a_i, a_j)$ is a similarity function that compares two words a_i and a_j using the symmetrized Tversky's index (Tversky, 1977; Jimenez et al., 2013a) representing them as sets of 3-grams of characters. That is, $a_i = \{a_{i,1}, a_{i,2}, ..., a_{i,|a_i|}\}$ where $a_{i,n}$ is the n^{th} character trigram in the word a_i in A. Thus, the proposed word-to-word similarity is given by:

$$sim(a_i, a_j) = \frac{|c|}{\beta(\alpha|a_{min}| + (1-\alpha)|a_{max}|) + |c|} \quad (2)$$

$$\begin{cases} |c| & = |a_i \cap a_j| + bias_{sim} \\ |a_{min}| & = \min\{|a_i \setminus a_j|, |a_j \setminus a_i|\} \\ |a_{max}| & = \max\{|a_i \setminus a_j|, |a_j \setminus a_i|\} \end{cases}$$

The sim function is equivalent to the Dice's coefficient if the three parameters are given their default values, namely $\alpha = 0.5$, $\beta = 1$ and $bias = 0$.

The soft cardinalities of any pair of sentences A, B and $A \cup B$ can be obtained using Eq. 1. The soft cardinality of the intersection is approximated by $|A \cap B|_{sim} = |A|_{sim} + |B|_{sim} - |A \cup B|_{sim}$. These four basic soft cardinalities are algebraically recombined to produce an extended set of 18 features as shown in Table 1. The feature $\mathbf{STS_{sim}}$ is a parameterized similarity function built by reusing at word level the symmetrized Tversky's index (Eq. 2), whose parameters are tuned from training data (as further described in Subsection 3.2).

Although this method is based purely on string matching, the soft cardinality has been shown to be a very strong baseline for semantic textual comparison. The word-to-word similarity sim in Eq. 1 could be replaced by other similarity functions based on semantic networks or any distributional representation making this method able to capture more complex semantic relations among words.

$\mathbf{STS_{sim}}$	$(\lvert A\rvert - \lvert A \cap B\rvert) / \lvert A\rvert$
$\lvert A\rvert$	$(\lvert A\rvert - \lvert A \cap B\rvert) / \lvert A \cup B\rvert$
$\lvert B\rvert$	$\lvert B\rvert / \lvert A \cup B\rvert$
$\lvert A \cap B\rvert$	$(\lvert B\rvert - \lvert A \cap B\rvert) / \lvert B\rvert$
$\lvert A \cup B\rvert$	$(\lvert B\rvert - \lvert A \cap B\rvert) / \lvert A \cup B\rvert$
$\lvert A\rvert - \lvert A \cap B\rvert$	$\lvert A \cap B\rvert / \lvert A\rvert$
$\lvert B\rvert - \lvert A \cap B\rvert$	$\lvert A \cap B\rvert / \lvert B\rvert$
$\lvert A \cup B\rvert - \lvert A \cap B\rvert$	$\lvert A \cap B\rvert / \lvert A \cup B\rvert$
$\lvert A\rvert / \lvert A \cup B\rvert$	$(\lvert A \cup B\rvert - \lvert A \cap B\rvert) / \lvert A \cap B\rvert$

NB: in this table only, $| * |$ is short for $| * |_{sim}$

Table 1: Soft cardinality features.

2.2 Sublexical Feature Representations

We have created a set of similarity measures based on induced representations of character n-grams. The measures are based on similarity between document vectors, here the centroid of the individual term vector representations, which are trained on character n-grams rather than full words. The vector representations are induced in an unsupervised manner from large unannotated corpora using word clustering, topic learning and word representation learning methods.

In this paper, three different methods have been used for creating the character n-gram feature representations: Brown Clusters (Brown et al., 1992), Latent Semantic Indexing (LSI) topics (Deerwester et al., 1990), and log linear skip-gram models (Mikolov et al., 2013). The Brown clusters were trained using the implementation by Liang (2005), while the LSI topic vectors and log linear skip-gram representations were trained using the Gensim topic modelling framework (Řehůřek and Sojka, 2010). In addition, *tf-idf* (Term-Frequency Inverse Document Frequency) weighting was used when training LSI topic models. We used a cosine distance measure between document vectors consisting of the centroid of the term representation vectors. For Brown clusters, the normalized term frequency vectors were used with the cluster IDs instead of the terms themselves. For LSI topic representations, the *tf-idf* weighted topic mixture for each term was used as the term representation. For the log linear skip-grams, the word representations were extracted from the model weight matrix.

3 Feature and Parameter Optimisation

The extracted features and the parameters for the two methods described in the previous section were optimised over several sets of training data. As no training data was explicitly provided for the STS evaluation campaign this year, we used different training sets from past campaigns and from Wikipedia for the new test sets.

Test set	Training set
deft-forum	MSRvid 2012 train and test + OnWN 2012 and 2013 test
deft-news	MSRvid 2012 train + test
headlines	headlines 2013 test
images	MSRvid 2012 train + test
OnWN	OnWN 2012 and 2013 test
tweet-news	SMTeuroparl 2012 test + SMTnews 2012 test

Table 2: Training-test set pairs.

3.1 Training Data and Pre-processing

The training-test sets pairs used for optimising the parameters of the soft cardinality methods were selected from the STS 2012 and STS 2013 task, as shown in Table 2. The character n-gram representation vectors were trained in an unsupervised manner on two subsets of Wikipedia consisting, respectively, of the first 12 million words (10^8 characters, hence referred to as *Wiki8*) and of 125 million words (10^9 characters; *Wiki9*).

First, however, the training data had to be preprocessed. Thus, before extracting the *idf* weights and the soft cardinality features, all the texts shown in Table 2 were passed through the following four pre-processing steps:

(i) tokenization and stop-word removal (provided by NLTK, Bird et al. (2009)),[1]

(ii) conversion to lowercase characters,

(iii) punctuation and special character removal (e.g., ".", ";", "$", "&"), and

(iv) Porter stemming.

Character n-grams including whitespace were generated from the Wikipedia texts, which in contrast only were pre-processed in a 3-step chain:

(i) removal of punctuation and extra whitespace,

(ii) replacing numbers with their single digit word ('one', 'two', etc.), and

(iii) lowercasing all text.

Data	α	β	$bias$	p	α'	β'	$bias'$
deft-forum	1.01	-1.01	0.24	0.93	-2.71	0.42	1.63
deft-news	3.36	-0.64	1.37	0.44	2.36	0.72	0.02
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
OnWN	0.53	-0.53	1.01	1.00	-4.89	0.52	0.46
tweet-news	0.13	0.14	2.80	0.01	2.66	1.74	0.45

Table 3: Optimal parameters used for each dataset.

3.2 Soft Cardinality Parameter Optimisation

The first feature in Table 1, $\mathbf{STS_{sim}}$, was used to optimise the four parameters α, β, *bias*, and p in the following way. First, we built a text similarity function reusing Eq. 2 for comparing two sets of words (instead of two sets of character 3-grams) and replacing the classic cardinality $|*|$ by the soft cardinality $|*|_{sim}$ from Eq. 1. This text similarity function adds three parameters (α', β', and $bias'$) to the initial four parameter set (α, β, *bias*, and p).

Second, these seven parameters were set to their default values and the scores obtained from this function for each pair of sentences were compared to the gold standards in the training data using Pearson's correlation. The parameter search space was then explored iteratively using hill-climbing until reaching optimal Pearson's correlation. The criterion for assignment of training-test set pairs was by closeness of average character length. The optimal training parameters are shown in Table 3.

3.3 Parameters for N-gram Feature Training

The character n-gram feature representation vectors were trained while varying the parameters of n-gram size, cluster size, and term frequency cut-offs for all models. For the log linear skip-gram models, our intuition is that a larger skip-gram context is needed than the 5 or 10 wide skip-grams used to train word-based representations due to the smaller term vocabulary and dependency between adjacent n-grams, so instead we trained models using skip-gram widths of 25 or 50 terms. Term frequency cut-offs were set to limit the model size, but also potentially serve as a regularization on the resulting measure. In detail, the following sublexical representation measures are used:

- Log linear skip-gram representations of character 3- and 4-grams of size 1000 and 2000, respectively. Trained on the Wiki8 corpus using a skip gram window of size 25 and 50, and frequency cut-off of 5.

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

- Brown clusters with size 1024 of character 4-grams using a frequency cut-off of 20.

- Brown clusters of character 3-, 4- and 5-grams with cluster sizes of resp. 1024, 2048 and 1024. The representations are trained on the Wiki9 corpus with successively increasing frequency cut-offs of 20, 320 and 1200.

- LSI topic vectors based on character 4-grams of size 2000. Trained on the Wiki8 corpus using a frequency cut-off of 5.

- LSI topic vectors based on character 4-grams of size 1000. Trained on the Wiki9 corpus using a frequency cut-off of 80.

3.4 Similarity Score Regression

The final sentence pair similarity score is predicted by a Support Vector Regression (SVR) model with a Radial Basis (RBF) kernel (Vapnik et al., 1997). The model is trained on all the test data for the 2013 STS shared task combined with all the trial and test data of the 2012 STS shared task.

The combined dataset hence consists of about 7,500 sentence pairs from nine different text categories: five sets from the annotated data supplied to STS 2012, based on Microsoft Research Paraphrase and Video description corpora (MSRpar and MSvid), statistical machine translation system output (SMTeuroparl and SMTnews), and sense mappings between OntoNotes and WordNet (OnWN); and four sets from the STS 2013 test data: headlines (news headlines), SMT, OnWN, and FNWM (mappings of sense definitions from FrameNet and WordNet).

The SVR model was trained as a bagged classifier, that is, for each run, 100 regression models were trained with 80% of the samples and features of the original training set drawn with replacement. The outputs of all models were then averaged into a final prediction. This bagged training procedure adds extra regularization, which can reduce the instability of prediction accuracy between different test data categories.

The prediction pipeline was implemented with the Scikit-learn software framework (Pedregosa et al., 2011), and the SVR models were trained with the implementation's default parameters: cost penalty (C) 1.0, margin (ϵ) 0.1, and RBF precision (γ) $1/|featurecount|$.

We were unable to improve the performance over these defaults by cross validation parameter search unless the models were trained for specific text categories. Consequently no parameter optimization was performed during training of the final systems.

4 Submitted Systems

The three submitted systems consist of one using only the soft cardinality features described in Section 3.2 (**NTNU-run1**), one system using a baseline set of lexical measures and WordNet augmented similarity in addition to the new sublexical representation measures (**NTNU-run2**), and one (**NTNU-run3**) which combines the output from the other two systems by taking the mean of the two sets of predictions. NTNU-run3 thus represents a combination of the measures and methods introduced by NTNU-run1 and NTNU-run2.

In addition to the sublexical feature measures described in Section 3.3, NTNU-run2 uses the following baseline features adapted from the TakeLab 2012 system submission (Šarić et al., 2012).

- Simple lexical features: Relative document length differences, number overlap, case overlap, and stock symbol named entity recognition.

- Lemma and word n-gram overlap of orders 1-3, frequency weighted lemma and word overlap, and WordNet augmented overlap.

- Cosine similarity between the summed word representation vectors from each sentence using LSI models based on large corpora with or without frequency weighting.

The specific measures used in the submitted systems were found by training the regression model on the STS 2012 shared task data and evaluating on the STS 2013 test data. We used a stepwise forward feature selection method by comparing mean (but unweighted) correlation on the four test categories in order to identify the subset of measures to include in the final system.

The system composes a feature set of similarity scores from these 20 baseline measures and the nine sublexical representation measures, and uses these to train a bagged SVM regressor as described in Section 3.4 in order to predict the final semantic similarity score for new sentence pairs.

	NTNU-run1		**NTNU-run2**		**NTNU-run3**		**Best**
Dataset	r	rank	r	rank	r	rank	r
deft-forum	0.4369	16	0.5084	2	0.5305	1	0.5305
deft-news	0.7138	14	0.7656	6	0.7813	2	0.7850
headlines	0.7219	17	0.7525	13	0.7837	1	0.7837
images	0.8000	9	0.8129	4	0.8343	1	0.8343
OnWN	0.8348	7	0.7767	20	0.8502	4	0.8745
tweet-news	0.4109	33	0.7921	1	0.6755	13	0.7921
mean	0.6531	20	0.7347	4	0.7426	2	0.7429
weighted mean	0.6631	21	0.7491	4	0.7549	3	0.7610

Table 4: Final evaluation results for the submitted systems.

5 Results and Discussion

The final evaluation results for the three submitted systems are shown in Table 4, where the rightmost column (‘Best’) for comparison displays the performance figures obtained by any of the 38 systems on each dataset.

The systems using sublexical representation based measures show competitive performance, ranking third and fourth among the submitted systems with a weighted mean correlation of $\sim$0.75. They also produced the best result in four out of the six text categories in the evaluation dataset, with NTNU-run3 being the #1 system on deft-forum, headlines and images, #2 on deft-news, and #4 on OnWN. It would thus have been the clear winner if it had not been for its sub-par performance on the tweet-news dataset, which on the other hand is the category NTNU-run2 was the best of all systems on.

The system based solely on soft cardinality features, NTNU-run1, displays more modest performance ranking at 21st place (of the in total 38 submitted systems) with $\sim$0.66 correlation. This is a bit surprising, since this method for obtaining features from pairs of texts was used successfully in other SemEval tasks such as cross-lingual textual entailment (Jimenez et al., 2012b) and student response analysis (Jimenez et al., 2013b). Similarly, Croce et al. (2012) used soft cardinality representing text as a bag of dependencies (syntactic soft cardinality) obtaining the best results in the typed-similarity task (Croce et al., 2013).

From our results it can be noted that for most categories the sublexical representation measures show strong performance in NTNU-run2, with a significantly better result for the combined system NTNU-run3. This indicates that while the soft cardinality features are weaker predictors overall, they are complimentary to the sublexical and lexical features of NTNU-run2. It is also indicative that this is not the case for the tweet-news category, where the text is more “free form” and less normative, so it would be expected that sublexical approaches should have stronger performance.

Acknowledgements

This work was made possible with the support from Department of Computer and Information Science, Norwegian University of Science and Technology.

Partha Pakray was 2013–2014 supported by an ERCIM Alain Bensoussan Fellowship.

The NTNU systems are partly based on code made available by the Text Analysis and Knowledge Engineering Laboratory, Department of Electronics, Microelectronics, Computer and Intelligent Systems, Faculty of Electrical Engineering and Computing, University of Zagreb.

References

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA, June.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media, Inc.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

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: Semantic Textual Similarity*, pages 59–65, Atlanta, Georgia, USA, June.

Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407.

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 *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, Montréal, Canada, 7-8 June.

Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. 2012b. Soft cardinality+ ML: Learning adaptive similarity functions for cross-lingual textual entailment. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, Montréal, Canada, 7-8 June.

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: Semantic Textual Similarity*, Atlanta, Georgia, USA, June.

Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. 2013b. SOFTCARDINALITY: Hierarchical text overlap for student response analysis. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, Atlanta, Georgia, USA, June.

Percy Liang. 2005. *Semi-supervised learning for natural language*. Ph.D. thesis, Massachusetts Institute of Technology.

Erwin Marsi, Hans Moen, Lars Bungum, Gleb Sizov, Björn Gambäck, and André Lynum. 2013. NTNU-CORE: Combining strong features for semantic similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 66–73, Atlanta, Georgia, USA, June.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA.

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

Vladimir Vapnik, Steven E. Golowich, and Alex Smola. 1997. Support vector method for function approximation, regression estimation, and signal processing. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 281–287. MIT Press, Cambridge, Massachusetts.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. TakeLab: Systems for measuring semantic text similarity. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 441–448, Montréal, Canada, 7-8 June.