LIPN-CORE: Semantic Text Similarity using n-grams, WordNet, Syntactic Analysis, ESA and Information Retrieval based Features

Davide Buscaldi, Joseph Le Roux,
Jorge J. García Flores
Laboratoire d'Informatique de Paris Nord,
CNRS, (UMR 7030)
Université Paris 13, Sorbonne Paris Cité,
F-93430, Villetaneuse, France
{buscaldi,joseph.le-roux,jgflores}
@lipn.univ-paris13.fr

Adrian Popescu
CEA, LIST,
Vision & Content
Engineering Laboratory
F-91190 Gif-sur-Yvette, France
adrian.popescu@cea.fr

Abstract

This paper describes the system used by the LIPN team in the Semantic Textual Similarity task at SemEval 2013. It uses a support vector regression model, combining different text similarity measures that constitute the features. These measures include simple distances like Levenshtein edit distance, cosine, Named Entities overlap and more complex distances like Explicit Semantic Analysis, WordNet-based similarity, IR-based similarity, and a similarity measure based on syntactic dependencies.

1 Introduction

The Semantic Textual Similarity task (STS) at SemEval 2013 requires systems to grade the degree of similarity between pairs of sentences. It is closely related to other well known tasks in NLP such as textual entailment, question answering or paraphrase detection. However, as noticed in (Bär et al., 2012), the major difference is that STS systems must give a *graded*, as opposed to binary, answer.

One of the most successful systems in SemEval 2012 STS, (Bär et al., 2012), managed to grade pairs of sentences accurately by combining focused measures, either simple ones based on surface features (*ie* n-grams), more elaborate ones based on lexical semantics, or measures requiring external corpora such as Explicit Semantic Analysis, into a robust measure by using a log-linear regression model.

The LIPN-CORE system is built upon this idea of combining simple measures with a regression model to obtain a robust and accurate measure of textual similarity, using the individual measures as features for the global system. These measures include simple distances like Levenshtein edit distance, cosine, Named Entities overlap and more complex distances like Explicit Semantic Analysis, WordNet-based similarity, IR-based similarity, and a similarity measure based on syntactic dependencies.

The paper is organized as follows. Measures are presented in Section 2. Then the regression model, based on Support Vector Machines, is described in Section 3. Finally we discuss the results of the system in Section 4.

2 Text Similarity Measures

2.1 WordNet-based Conceptual Similarity (Proxigenea)

First of all, sentences p and q are analysed in order to extract all the included WordNet synsets. For each WordNet synset, we keep noun synsets and put into the set of synsets associated to the sentence, C_p and C_q, respectively. If the synsets are in one of the other POS categories (verb, adjective, adverb) we look for their derivationally related forms in order to find a related noun synset: if there is one, we put this synsets in C_p (or C_q). For instance, the word "playing" can be associated in WordNet to synset `(v)play#2`, which has two derivationally related forms corresponding to synsets `(n)play#5` and `(n)play#6`: these are the synsets that are added to the synset set of the sentence. No disambiguation process is carried out, so we take all possible meanings into account.

Given C_p and C_q as the sets of concepts contained in sentences p and q, respectively, with $|C_p| \geq |C_q|$, the conceptual similarity between p and q is calculated as:

$$ss(p,q) = \frac{\sum\limits_{c_1 \in C_p} \max\limits_{c_2 \in C_q} s(c_1, c_2)}{|C_p|} \tag{1}$$

where $s(c_1, c_2)$ is a conceptual similarity measure. Concept similarity can be calculated by different ways. For the participation in the 2013 Semantic Textual Similarity task, we used a variation of the Wu-Palmer formula (Wu and Palmer, 1994) named "ProxiGenea" (from the french Proximité Généalogique, genealogical proximity), introduced by (Dudognon et al., 2010), which is inspired by the analogy between a family tree and the concept hierarchy in WordNet. Among the different formulations proposed by (Dudognon et al., 2010), we chose the ProxiGenea3 variant, already used in the STS 2012 task by the IRIT team (Buscaldi et al., 2012). The ProxiGenea3 measure is defined as:

$$s(c_1, c_2) = \frac{1}{1 + d(c_1) + d(c_2) - 2 \cdot d(c_0)} \tag{2}$$

where c_0 is the most specific concept that is present both in the synset path of c_1 and c_2 (that is, the Least Common Subsumer or LCS). The function returning the depth of a concept is noted with d.

2.2 IC-based Similarity

This measure has been proposed by (Mihalcea et al., 2006) as a corpus-based measure which uses Resnik's Information Content (IC) and the Jiang-Conrath (Jiang and Conrath, 1997) similarity metric:

$$s_{jc}(c_1, c_2) = \frac{1}{IC(c_1) + IC(c_2) - 2 \cdot IC(c_0)} \tag{3}$$

where IC is the information content introduced by (Resnik, 1995) as $IC(c) = -\log P(c)$.

The similarity between two text segments T_1 and T_2 is therefore determined as:

$$\begin{aligned} sim(T_1, T_2) &= \frac{1}{2} \left(\frac{\sum\limits_{w \in \{T_1\}} \max\limits_{w_2 \in \{T_2\}} ws(w, w_2) * idf(w)}{\sum\limits_{w \in \{T_1\}} idf(w)} \right. \\ &+ \left. \frac{\sum\limits_{w \in \{T_2\}} \max\limits_{w_1 \in \{T_1\}} ws(w, w_1) * idf(w)}{\sum\limits_{w \in \{T_2\}} idf(w)} \right) \end{aligned} \tag{4}$$

where $idf(w)$ is calculated as the inverse document frequency of word w, taking into account Google Web 1T (Brants and Franz, 2006) frequency counts. The semantic similarity between words is calculated as:

$$ws(w_i, w_j) = \max_{c_i \in W_i, c_j in W_j} s_{jc}(c_i, c_j). \tag{5}$$

where W_i and W_j are the sets containing all synsets in WordNet corresponding to word w_i and w_j, respectively. The IC values used are those calculated by Ted Pedersen (Pedersen et al., 2004) on the British National Corpus[1].

2.3 Syntactic Dependencies

We also wanted for our systems to take syntactic similarity into account. As our measures are lexically grounded, we chose to use dependencies rather than constituents. Previous experiments showed that converting constituents to dependencies still achieved best results on out-of-domain texts (Le Roux et al., 2012), so we decided to use a 2-step architecture to obtain syntactic dependencies. First we parsed pairs of sentences with the LORG parser[2]. Second we converted the resulting parse trees to Stanford dependencies[3].

Given the sets of parsed dependencies D_p and D_q, for sentence p and q, a dependency $d \in D_x$ is a triple (l, h, t) where l is the dependency label (for instance, *dobj* or *prep*), h the governor and t the dependant. We define the following similarity measure between two syntactic dependencies $d_1 = (l_1, h_1, t_1)$ and $d_2 = (l_2, h_2, t_2)$:

$$\begin{aligned} dsim(d_1, d_2) &= Lev(l_1, l_2) \\ &* \frac{idf_h * s_{WN}(h_1, h_2) + idf_t * s_{WN}(t_1, t_2)}{2} \end{aligned} \tag{6}$$

where $idf_h = \max(idf(h_1), idf(h_2))$ and $idf_t = \max(idf(t_1), idf(t_2))$ are the inverse document frequencies calculated on Google Web 1T for the governors and the dependants (we retain the maximum for each pair), and s_{WN} is calculated using formula 2, with two differences:

- if the two words to be compared are antonyms, then the returned score is 0;

[1] http://www.d.umn.edu/~tpederse/similarity.html

[2] https://github.com/CNGLdlab/LORG-Release

[3] We used the default built-in converter provided with the Stanford Parser (2012-11-12 revision).

- if one of the words to be compared is not in WordNet, their similarity is calculated using the Levenshtein distance.

The similarity score between p and q, is then calculated as:

$$s_{SD}(p,q) = \max\left(\frac{\sum\limits_{d_i \in D_p} \max\limits_{d_j in D_q} dsim(d_i,d_j)}{|D_p|}, \frac{\sum\limits_{d_i \in D_q} \max\limits_{d_j in D_p} dsim(d_i,d_j)}{|D_q|}\right) \tag{7}$$

2.4 Information Retrieval-based Similarity

Let us consider two texts p and q, an Information Retrieval (IR) system S and a document collection D indexed by S. This measure is based on the assumption that p and q are similar if the documents retrieved by S for the two texts, used as input queries, are ranked similarly.

Let be $L_p = \{d_{p_1}, \ldots, d_{p_K}\}$ and $L_q = \{d_{q_1}, \ldots, d_{q_K}\}$, $d_{x_i} \in D$ the sets of the top K documents retrieved by S for texts p and q, respectively. Let us define $s_p(d)$ and $s_q(d)$ the scores assigned by S to a document d for the query p and q, respectively. Then, the similarity score is calculated as:

$$sim_{IR}(p,q) = 1 - \frac{\sum\limits_{d \in L_p \cap L_q} \frac{\sqrt{(s_p(d)-s_q(d))^2}}{\max(s_p(d),s_q(d))}}{|L_p \cap L_q|} \tag{8}$$

if $|L_p \cap L_q| \neq \emptyset$, 0 otherwise.

For the participation in this task we indexed a collection composed by the AQUAINT-2[4] and the English NTCIR-8[5] document collections, using the Lucene[6] 4.2 search engine with BM25 similarity. The K value was empirically set to 20 after some tests on the SemEval 2012 data.

2.5 ESA

Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007) represents meaning as a weighted vector of Wikipedia concepts. Weights are supposed to quantify the strength of the relation between a word and each Wikipedia concept using the *tf-idf* measure. A text is then represented as a high-dimensional real valued vector space spanning all along the Wikipedia database. For this particular task we adapt the *research-esa* implementation (Sorg and Cimiano, 2008)[7] to our own home-made weighted vectors corresponding to a Wikipedia snapshot of February 4th, 2013.

2.6 N-gram based Similarity

This feature is based on the Clustered Keywords Positional Distance (CKPD) model proposed in (Buscaldi et al., 2009) for the passage retrieval task.

The similarity between a text fragment p and another text fragment q is calculated as:

$$sim_{ngrams}(p,q) = \frac{\sum\limits_{\forall x \in Q} h(x,P)\frac{1}{d(x,x_{max})}}{\sum_{i=1}^{n} w_i} \tag{9}$$

Where P is the set of *n*-grams with the highest weight in p, where all terms are also contained in q; Q is the set of all the possible n-grams in q and n is the total number of terms in the longest passage. The weights for each term and each n-gram are calculated as:

- w_i calculates the weight of the term t_I as:

  $$w_i = 1 - \frac{log(n_i)}{1 + log(N)} \tag{10}$$

  Where n_i is the frequency of term t_i in the Google Web 1T collection, and N is the frequency of the most frequent term in the Google Web 1T collection.

- the function $h(x,P)$ measures the weight of each *n*-gram and is defined as:

  $$h(x,P_j) = \begin{cases} \sum_{k=1}^{j} w_k & \text{if } x \in P_j \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

[4] http://www.nist.gov/tac/data/data_desc.html#AQUAINT-2

[5] http://metadata.berkeley.edu/NTCIR-GeoTime/ntcir-8-databases.php

[6] http://lucene.apache.org/core

[7] http://code.google.com/p/research-esa/

Where w_k is the weight of the *k-th* term (see Equation 10) and *j* is the number of terms that compose the n-gram x;

- $\frac{1}{d(x,x_{max})}$ is a distance factor which reduces the weight of the *n*-grams that are far from the heaviest *n*-gram. The function $d(x, x_{max})$ determines numerically the value of the separation according to the number of words between a *n*-gram and the heaviest one:

$$d(x, x_{max}) = 1 + k \cdot ln(1 + L) \tag{12}$$

where *k* is a factor that determines the importance of the distance in the similarity calculation and *L* is the number of words between a *n*-gram and the heaviest one (see Equation 11). In our experiments, k was set to 0.1, the default value in the original model.

2.7 Other measures

In addition to the above text similarity measures, we used also the following common measures:

2.7.1 Cosine

Given $\mathbf{p} = (w_{p_1}, \ldots, w_{p_n})$ and $\mathbf{q} = (w_{q_1}, \ldots, w_{q_n})$ the vectors of $tf.idf$ weights associated to sentences p and q, the cosine distance is calculated as:

$$sim_{cos}(\mathbf{p}, \mathbf{q}) = \frac{\sum\limits_{i=1}^{n} w_{p_i} \times w_{q_i}}{\sqrt{\sum\limits_{i=1}^{n} {w_{p_i}}^2} \times \sqrt{\sum\limits_{i=1}^{n} {w_{q_i}}^2}} \tag{13}$$

The idf value was calculated on Google Web 1T.

2.7.2 Edit Distance

This similarity measure is calculated using the Levenshtein distance as:

$$sim_{ED}(p, q) = 1 - \frac{Lev(p, q)}{\max(|p|, |q|)} \tag{14}$$

where $Lev(p, q)$ is the Levenshtein distance between the two sentences, taking into account the characters.

2.7.3 Named Entity Overlap

We used the Stanford Named Entity Recognizer by (Finkel et al., 2005), with the 7 class model trained for MUC: Time, Location, Organization, Person, Money, Percent, Date. Then we calculated a per-class overlap measure (in this way, “France” as an Organization does not match “France” as a Location):

$$O_{NER}(p, q) = \frac{2 * |N_p \cap N_q|}{|N_p| + |N_q|} \tag{15}$$

where N_p and N_q are the sets of NEs found, respectively, in sentences p and q.

3 Integration of Similarity Measures

The integration has been carried out using the ν-Support Vector Regression model (ν-SVR) (Schölkopf et al., 1999) implementation provided by LIBSVM (Chang and Lin, 2011), with a radial basis function kernel with the standard parameters ($\nu = 0.5$).

4 Results

In order to evaluate the impact of the different features, we carried out an ablation test, removing one feature at a time and training a new model with the reduced set of features. In Table 2 we show the results of the ablation test for each subset of the SemEval 2013 test set; in Table 1 we show the same test on the whole test set. Note: the results have been calculated as the Pearson correlation test on the whole test set and not as an average of the correlation scores calculated over the composing test sets.

Feature Removed	Pearson	Loss
None	0.597	0
N-grams	0.596	0.10%
WordNet	0.563	3.39%
SyntDeps	0.602	−0.43%
Edit	0.584	1.31%
Cosine	0.596	0.10%
NE Overlap	0.603	−0.53%
IC-based	0.598	−0.10%
IR-Similarity	0.510	**8.78%**
ESA	0.601	−0.38%

Table 1: Ablation test for the different features on the whole 2013 test set.

	FNWN		**Headlines**		**OnWN**		**SMT**	
Feature Removed	**Pearson**	**Loss**	**Pearson**	**Loss**	**Pearson**	**Loss**	**Pearson**	**Loss**
None	0.404	0	0.706	0	0.694	0	0.301	0
N-grams	0.379	2.49%	0.705	0.12%	0.698	−0.44%	0.289	1.16%
WordNet	0.376	**2.80%**	0.695	1.09%	0.682	1.17%	0.278	2.28%
SyntDeps	0.403	0.08%	0.699	0.70%	0.679	1.49%	0.284	1.62%
Edit	0.402	0.19%	0.689	1.70%	0.667	2.72%	0.286	1.50%
Cosine	0.393	1.03%	0.683	2.38%	0.676	1.80%	0.303	−0.24%
NE Overlap	0.410	−0.61%	0.700	0.67%	0.680	1.37%	0.285	1.58%
IC-based	0.391	1.26%	0.699	0.75%	0.669	2.50%	0.283	1.76%
IR-Similarity	0.426	−2.21%	0.633	**7.33%**	0.589	**10.46%**	0.249	**5.19%**
ESA	0.391	1.22%	0.691	1.57%	0.702	−0.81%	0.275	2.54%

Table 2: Ablation test for the different features on the different parts of the 2013 test set.

	FNWN	**Headlines**	**OnWN**	**SMT**	**ALL**
N-grams	0.285	0.532	0.459	0.280	0.336
WordNet	0.395	**0.606**	0.552	0.282	0.477
SyntDeps	0.233	0.409	0.345	0.323	0.295
Edit	0.220	0.536	0.089	**0.355**	0.230
Cosine	0.306	0.573	0.541	0.244	0.382
NE Overlap	0.000	0.216	0.000	0.013	0.020
IC-based	**0.413**	0.540	**0.642**	0.285	0.421
IR-based	0.067	0.598	0.628	0.241	**0.541**
ESA	0.328	0.546	0.322	0.289	0.390

Table 3: Pearson correlation calculated on individual features.

The ablation test show that the IR-based feature showed up to be the most effective one, especially for the headlines subset (as expected), and, quite surprisingly, on the OnWN data. In Table 3 we show the correlation between each feature and the result (feature values normalised between 0 and 5): from this table we can also observe that, on average, IR-based similarity was better able to capture the semantic similarity between texts. The only exception was the FNWN test set: the IR-based similarity returned a 0 score 178 times out of 189 (94.1%), indicating that the indexed corpus did not fit the content of the FNWN sentences. This result shows also the limits of the IR-based similarity score which needs a large corpus to achieve enough coverage.

4.1 Shared submission with INAOE-UPV

One of the files submitted by INAOE-UPV, `INAOE-UPV-run3` has been produced using seven features produced by different teams: INAOE, LIPN and UMCC-DLSI. We contributed to this joint submission with the IR-based, WordNet and cosine features.

5 Conclusions and Further Work

In this paper we introduced the LIPN-CORE system, which combines semantic, syntactic an lexical measures of text similarity in a linear regression model. Our system was among the best 15 runs for the STS task. According to the ablation test, the best performing feature was the IR-based one, where a sentence is considered as a query and its meaning represented as a set of documents indexed by an IR system. The second and third best-performing measures were WordNet similarity and Levenshtein's edit distance. On the other hand, worst performing similarity measures were Named Entity Overlap, Syntactic Dependencies and ESA. However, a correlation analysis calculated on the features taken one-by-one shows that the contribution of a feature on the overall regression result does not correspond to the actual capability of the measure to represent the semantic similarity between the two texts. These results raise the methodological question of how to combine semantic, syntactic and lexical similarity measures in order to estimate the impact of the different strategies used on each dataset.

Further work will include richer similarity measures, like quasi-synchronous grammars (Smith and Eisner, 2006) and random walks (Ramage et al., 2009). Quasi-synchronous grammars have been used successfully for paraphrase detection (Das and Smith, 2009), as they provide a fine-grained modeling of the alignment of syntactic structures, in a very flexible way, enabling partial alignments and the inclusion of external features, like Wordnet lexical relations for example. Random walks have been used effectively for paraphrase recognition and as a feature for recognizing textual entailment. Finally, we will continue analyzing the question of how to combine a wide variety of similarity measures in such a way that they tackle the semantic variations of each dataset.

References

[Bär et al.2012] 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, held in conjunction with the 1st Joint Conference on Lexical and Computational Semantics*, pages 435–440, Montreal, Canada, June.

[Brants and Franz2006] Thorsten Brants and Alex Franz. 2006. Web 1t 5-gram corpus version 1.1.

[Buscaldi et al.2009] Davide Buscaldi, Paolo Rosso, José Manuel Gómez, and Emilio Sanchis. 2009. Answering questions with an n-gram based passage retrieval engine. *Journal of Intelligent Information Systems (JIIS)*, 34(2):113–134.

[Buscaldi et al.2012] Davide Buscaldi, Ronan Tournier, Nathalie Aussenac-Gilles, and Josiane Mothe. 2012. Irit: Textual similarity combining conceptual similarity with an n-gram comparison method. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012)*, Montreal, Quebec, Canada.

[Chang and Lin2011] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Das and Smith2009] Dipanjan Das and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proc. of ACL-IJCNLP*.

[Dudognon et al.2010] Damien Dudognon, Gilles Hubert, and Bachelin Jhonn Victorino Ralalason. 2010. Proxigénéa : Une mesure de similarité conceptuelle. In *Proceedings of the Colloque Veille Stratégique Scientifique et Technologique (VSST 2010)*.

[Finkel et al.2005] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Gabrilovich and Markovitch2007] 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, pages 1606–1611, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Jiang and Conrath1997] J.J. Jiang and D.W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the Int'l. Conf. on Research in Computational Linguistics*, pages 19–33.

[Le Roux et al.2012] Joseph Le Roux, Jennifer Foster, Joachim Wagner, Rasul Samad Zadeh Kaljahi, and Anton Bryl. 2012. DCU-Paris13 Systems for the SANCL 2012 Shared Task. In *The NAACL 2012 First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*, pages 1–4, Montréal, Canada, June.

[Mihalcea et al.2006] Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 1*, AAAI'06, pages 775–780. AAAI Press.

[Pedersen et al.2004] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL–Demonstrations '04, pages 38–41, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Ramage et al.2009] Daniel Ramage, Anna N. Rafferty, and Christopher D. Manning. 2009. Random walks for text semantic similarity. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural*

Language Processing, pages 23–31. The Association for Computer Linguistics.

[Resnik1995] Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Schölkopf et al.1999] Bernhard Schölkopf, Peter Bartlett, Alex Smola, and Robert Williamson. 1999. Shrinking the tube: a new support vector regression algorithm. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 330–336, Cambridge, MA, USA. MIT Press.

[Smith and Eisner2006] David A. Smith and Jason Eisner. 2006. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proceedings of the HLT-NAACL Workshop on Statistical Machine Translation*, pages 23–30, New York, June.

[Sorg and Cimiano2008] Philipp Sorg and Philipp Cimiano. 2008. Cross-lingual Information Retrieval with Explicit Semantic Analysis. In *Working Notes for the CLEF 2008 Workshop*.

[Wu and Palmer1994] Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.