BUAP: Three Approaches for Semantic Textual Similarity

Maya Carrillo, Darnes Vilariño, David Pinto, Mireya Tovar, Saul León, Esteban Castillo
Benemérita Universidad Autónoma de Puebla,
Faculty of Computer Science
14 Sur & Av. San Claudio, CU
Puebla, Puebla, México
{cmaya, darnes, dpinto, mtovar}@cs.buap.mx
saul.ls@live.com, ecjbuap@gmail.com

Abstract

In this paper we describe the three approaches we submitted to the Semantic Textual Similarity task of SemEval 2012. The first approach considers to calculate the semantic similarity by using the Jaccard coefficient with term expansion using synonyms. The second approach uses the semantic similarity reported by Mihalcea in (Mihalcea et al., 2006). The third approach employs Random Indexing and Bag of Concepts based on context vectors. We consider that the first and third approaches obtained a comparable performance, meanwhile the second approach got a very poor behavior. The best ALL result was obtained with the third approach, with a Pearson correlation equal to 0.663.

1 Introduction

Finding the semantic similarity between two sentences is very important in applications of natural language processing such as information retrieval and related areas. The problem is complex due to the small number of terms involved in sentences which are tipically less than 10 or 15. Additionally, it is required to "understand" the meaning of the sentences in order to determine the "semantic" similarity of texts, which is quite different of finding the lexical similarity.

There exist different works at literature dealing with semantic similarity, but the problem is far to be solved because of the aforementioned issues. In (Mihalcea et al., 2006), for instance, it is presented a method for measuring the semantic similarity of texts, using corpus-based and knowledge-based measures of similarity. The approaches presented in (Shrestha, 2011) are based on the Vector Space Model, with the aim to capture the contextual behavior, senses and correlation, of terms. The performance of the method is better than the baseline method that uses vector based cosine similarity measure.

In this paper, we present three different approaches for the Textual Semantic Similarity task of Semeval 2012 (Agirre et al., 2012). The task is described as follows: Given two sentences s_1 and s_2, the aim is to compute how similar s_1 and s_2 are, returning a similarity score, and an optional confidence score. The approaches should provide values between 0 and 5 for each pair of sentences. These values roughly correspond to the following considerations, even when the system should output real values:

5: The two sentences are completely equivalent, as they mean the same thing.

4: The two sentences are mostly equivalent, but some unimportant details differ.

3: The two sentences are roughly equivalent, but some important information differs/missing.

2: The two sentences are not equivalent, but share some details.

1: The two sentences are not equivalent, but are on the same topic.

0: The two sentences are on different topics.

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 631–634,
Montréal, Canada, June 7-8, 2012.

The description of the runs submitted to the competition follows.

2 Experimentation setup

The three runs submitted to the competition use completely different mechanisms to find the degree of semantic similarity between two sentences. The approaches are described as follows:

2.1 Approach BUAP-RUN-1: Term expansion with synonyms

Let $s_1 = w_{1,1}w_{1,2}...w_{1,|s_1|}$ and $s_2 = w_{2,1}w_{2,2}...w_{2,|s_2|}$ be two sentences. The synonyms of a given word $w_{i,k}$, expressed as $synonyms(w_{i,k})$, are obtained from online dictionaries by extracting the synonyms of $w_{i,k}$. A better matching between the terms contained in the text fragments and the terms at the dictionary are obtained by stemming all the terms (using the Porter stemmer).

In order to determine the semantic similarity between any pair of terms of the two sentences ($w_{1,i}$ and $w_{2,j}$) we use Eq. (1).

$$sim(w_{1,i}, w_{2,j}) = \begin{cases} 1 & \text{if } (w_{1,i} == w_{2,j}) \;||\\ & w_{1,i} \in synonyms(w_{2,j}) \;||\\ & w_{2,j} \in synonyms(w_{1,i}) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The similarity between sentences s_1 and s_2 is calculated as shown in Eq. (2).

$$similarity(s_1, s_2) = \frac{5 * \sum_{i=1}^{n} \sum_{j=1}^{n} sim(w_{1i}, w_{2j})}{|s_1 \cup s_2|} \quad (2)$$

2.2 Approach BUAP-RUN-2

In this approach, the similarity of s_1 and s_2 is calculated as shown in Eq. (3) (Mihalcea et al., 2006).

$$similarity(s_1, s_2) = \frac{1}{2}\left(\frac{\sum_{w \in \{s_1\}} (maxSim(w, s_2) * idf(w))}{\sum_{w \in \{s_1\}} idf(w)} + \frac{\sum_{w \in \{s_2\}} (maxSim(w, s_1) * idf(w))}{\sum_{w \in \{s_2\}} idf(w)}\right) \quad (3)$$

where $idf(w)$ is the inverse document frequency of the word w, and $maxSim(w, s_2)$ is the maximum lexical similarity between the word w in sentence s_2 and all the words in sentence s_2 calculated by means of the Eq. (4) reported by (Wu and Palmer, 1994). The sentence terms are assumed to be concepts, LCS is the depth of the least common subsumer, and the equation is calculated using the NLTK libraries[1].

$$Sim_{wup} = \frac{2 * depth(LCS)}{depth(concept_1) + depth(concept_2)} \quad (4)$$

2.3 Approach BUAP-RUN-3: Random Indexing and Bag of Concepts

The vector space model (VSM) for document representation supporting search is probably the most well-known IR model. The VSM assumes that term vectors are pair-wise orthogonal. This assumption is very restrictive because words are not independent. There have been various attempts to build representations for documents that are semantically richer than only vectors based on the frequency of terms occurrence. One example is Latent Semantic Indexing (LSI), a method of word co-occurrence analysis to compute semantic vectors (context vectors) for words. LSI applies singular-value decomposition (SVD) to the term-document matrix in order to construct context vectors. As a result the dimension of the produced vector space will be significantly smaller; consequently the vectors that represent terms cannot be orthogonal. However, dimension reduction techniques such as SVD are expensive in terms of memory and processing time. Performing the SVD takes time *O (nmz)*, where *n* is the vocabulary size, *m* is the number of documents, and *z* is the number of nonzero elements per column in the words-by-documents matrix. As an alternative, there is a vector space methodology called Random Indexing (RI) (Sahlgren, 2005), which presents an efficient, scalable, and incremental method for building context vectors. Its computational complexity is *O (nr)* where *n* is as previously described and *r* is the vector dimension. Particularly, we apply RI to capture the inherent semantic structure using Bag of Concepts representation (BoC) as proposed by Sahlgren and Cöster (Sahlgren and Cöster, 2004), where the meaning of a term is considered as the sum of contexts in which it occurs.

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

2.3.1 Random Indexing

Random Indexing (RI) is a vector space methodology that accumulates context vectors for words based on co-occurrence data. The technique can be described as:

- First a unique random representation known as index vector is assigned to each context (document). Index vectors are binary vectors with a small number of non-zero elements, which are either +1 or -1, with equal amounts of both. For example, if the index vectors have twenty non-zero elements in a 1024-dimensional vector space, they have ten +1s and ten -1s. Index vectors serve as indices or labels for documents

- Index vectors are used to produce context vectors by scanning through the text and every time a target word occurs in a context, the index vector of the context is added to the context vector of the target word. Thus, at each encounters of the target word *t* with a context *c* the context vector of *t* is updated as follows: *ct* + = *ic* where *ct* is the context vector of *t* and *ic* is the index vector of *c*. In this way, the context vector of a word keeps track of the contexts in which it occurred.

RI methodology is similar to latent semantic indexing (LSI) (Deerwester et al., 1990). However, to reduce the co-occurrence matrix no dimension reduction technique such as SVD is needed, since the dimensionality *d* of the random index vectors is preestablished as a parameter (implicit dimension reduction). Consequently *d* does not change once it has been set; as a result, the dimensionality of context vectors will never change with the addition of new data.

2.3.2 Bag of Concepts

Bag of Concepts (BoC) is a recent representation scheme proposed by Sahlgren and Cöster in (Sahlgren and Cöster, 2004), which is based on the perception that the meaning of a document can be considered as the union of the meanings of its terms. This is accomplished by generating term context vectors from each term within the document, and generating a document vector as the weighted sum of the term context vectors contained within that document. Therefore, we use RI to represent the meaning of a word as the sum of contexts (entire documents) in which it occurs. Illustrating this technique, suppose you have two documents: *D1: A man with a hard hat is dancing*, and *D2: A man wearing a hard hat is dancing*. Let us suppose that they have index vectors *ID1* and *ID2*, respectively: the context vector for *hat* will be the *ID1 + ID2*, because this word appears in both documents. Once the context vectors have been built by RI, they are used to represent the document as BoC. For instance, supposing *CV1, CV2, CV3, ...* and *CV8*, are the context vectors of each word in *D1*, then document *D1* will be represented as the weighted sum of these eight context vectors.

2.3.3 Implementation

The sentences of each file were processed to generate the BoC representations of them. BoC representations were generated by first stemming all words in the sentences. We then used random indexing to produce context vectors for each word in the files (i.e. STS.input.MSRpar, STS.input.MSRvid, etc.), each file was considered a different corpus and documents were the sentences in them. The dimension of the context vectors was fixed at 2048, determined by experimentation using the training set. These context vectors were then $tf \times idf$-weighted, according to the corpus, and added up for each sentence, to produce BoC representations. Therefore the similarity values were calculated by the cosine function. Finally cosine values were multiplied by 5 to produce values between 0 and 5.

3 Experimental results

In Table 1 we show the results obtained by the three approaches submitted to the competition. The columns of Table 1 stand for:

- **ALL**: Pearson correlation with the gold standard for the five datasets, and corresponding rank.

- **ALLnrm**: Pearson correlation after the system outputs for each dataset are fitted to the gold standard using least squares, and corresponding rank.

Run	ALL	Rank	ALL nrm	Rank Nrm	Mean	Rank Mean	MSR par	MSR vid	SMT eur	On - WN	SMT-news
BUAP-RUN-1	0.4997	63	0.7568	62	0.4892	57	0.4037	0.6532	0.4521	0.605	0.4537
BUAP-RUN-2	-0.026	89	0.5933	89	0.0669	89	0.1109	0.0057	0.0348	0.1788	0.1964
BUAP-RUN-3	0.663	25	0.7474	64	0.488	59	0.4018	0.6378	0.4758	0.5691	0.4057

Table 1: Results of approaches of BUAP in Task 6.

- **Mean**: Weighted mean across the 5 datasets, where the weight depends on the number of pairs in the dataset.

Followed by Pearson for individual datasets.

At this moment, we are not aware of the reasons because the second approach obtained a very poor performance. The way in which the $idf(w)$ is calculated could be one of the reasons, because the corpus used is relatively small and also from a different domain. With respect to the other two approaches, we consider that they (first and third) obtained a comparable performance, even when the third approach obtained the best ALL result with a Pearson correlation equal to 0.663.

4 Discussion and conclusion

We have presented three different approaches for tackling the problem of Semantic Textual Similarity. The use of term expansion by synonyms performed well in general and obtained a comparable behavior than the third approach which used random indexing and bag of concepts. It is interesting to observe that these two approaches performed similar when the two term expansion mechanism are totally different. As further, it is important to analyze the poor behavior of the second approach. We would like also to introduce semantic relationships other than synonyms in the process of term expansion.

Acknowledgments

This project has been partially supported by projects CONACYT #106625, #VIAD-ING11-II, PROMEP/103.5/11/4481 and VIEP #PIAD-ING11-II.

References

E. Agirre, D. Cer, M. Diab, and B. Dolan. 2012. SemEval-2012 Task 6: Semantic Textual Similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012)*.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

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

Magnus Sahlgren and Rickard Cöster. 2004. Using bag-of-concepts to improve the performance of support vector machines in text categorization. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

M. Sahlgren. 2005. An Introduction to Random Indexing. *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005*.

Prajol Shrestha. 2011. Corpus-based methods for short text similarity. In *TALN 2011*, Montpellier, France.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133 –138, New Mexico State University, Las Cruces, New Mexico.