IBM_EG-CORE: Comparing multiple Lexical and NE matching features in measuring Semantic Textual similarity

Sara Noeman
IBM Cairo Technology and Development Center
Giza, Egypt
P.O. Box 166 Al-Ahram
noemans@eg.ibm.com

Abstract

We present in this paper the systems we participated with in the Semantic Textual Similarity task at SEM 2013. The Semantic Textual Similarity Core task (STS) computes the degree of semantic equivalence between two sentences where the participant systems will be compared to the manual scores, which range from 5 (semantic equivalence) to 0 (no relation). We combined multiple text similarity measures of varying complexity. The experiments illustrate the different effect of four feature types including direct lexical matching, idf-weighted lexical matching, modified BLEU N-gram matching and named entities matching. Our team submitted three runs during the task evaluation period and they ranked number 11, 15 and 19 among the 90 participating systems according to the official Mean Pearson correlation metric for the task. We also report an unofficial run with mean Pearson correlation of 0.59221 on STS2013 test dataset, ranking as the 3rd best system among the 90 participating systems.

1 Introduction

The Semantic Textual Similarity (STS) task at SEM 2013 is to measure the degree of semantic equivalence between pairs of sentences as a graded notion of similarity. Text Similarity is very important to many Natural Language Processing applications, like extractive summarization (Salton et al., 1997), methods for automatic evaluation of machine translation (Papineni et al., 2002), as well as text summarization (Lin and Hovy, 2003). In Text Coherence Detection (Lapata and Barzilay, 2005), sentences are linked together by similar or related words. For Word Sense Disambiguation, researchers (Banerjee and Pedersen, 2003; Guo and Diab, 2012a) introduced a sense similarity measure using the sentence similarity of the sense definitions. In this paper we illustrate the different effect of four feature types including direct lexical matching, idf-weighted lexical matching, modified BLEU N-gram matching and named entities matching. The rest of this paper will proceed as follows, Section 2 describes the four text similarity features used. Section 3 illustrates the system description, data resources as well as Feature combination. Experiments and Results are illustrated in section 4. then we report our conclusion and future work.

2 Text Similarity Features

Our system measures the semantic textual similarity between two sentences through a number of matching features which should cover four main dimensions: i) Lexical Matching ii) IDF-weighted Lexical Matching iii) Contextual sequence Matching (Modified BLEU Score), and iv) Named Entities Matching.

First we introduce the alignment technique used. For a sentence pair {s1, s2} matching is done in each direction separately to detect the sub-sentence of s1 matched to s2 and then detect the sub-sentence of s2 matched to s1. For each word *wi* in s1 we search for its match *wj* in s2 according to matching features.

S1: w0 w1 w2 w3 w4 …... wi …... wn
S2: w0 w1 w2 w3 w4 ….....wj …..... wm

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task*, pages 187–193, Atlanta, Georgia, June 13-14, 2013.

2.1 Lexical Matching:

In this feature we handle the two sentences as bags of words to be matched using three types of matching, given that all stop words are cleaned out before matching:

I) Exact word matching.

II) Stemmed word matching: I used Porter Stemming algorithm (M.F. Porter, 1980) in matching, where it is a process for removing the commoner morphological and inflectional endings from words in English. Stemming will render inflections like "requires, required, requirements, ..." to "requir" so they can be easily matched

III) Synonyms matching: we used a corpus based dictionary of 58,921 entries and their equivalent synonyms. The next section describes how we automatically generated this language resource.

2.2 IDF-weighted Lexical Matching

We used the three matching criteria used in *Lexical Matching* after weighting them with Inverse-Document-Frequency. we applied the aggregation strategy by Mihalcea et al. (2006): The sum of the idf-weighted similarity scores of each word with the best-matching counterpart in the other text is computed in both directions. For a sentence pair s1, s2, if s1 consists of m words {w_0, w_1, …., $w_{(m-1)}$} and s2 consists of n words {w_0, w_1, …., $w_{(n-1)}$} ,after cleaning stop words from both, and the matched words are "@Matched_word_List" of "k" words, then

$$Similarity(S1, S2) = \frac{\sum_{w \in Match\,S1\,S2} idf(w)}{\sum_{w \in S1} idf(w)};$$

$$Similarity(S2, S1) = \frac{\sum_{w \in Match\,S1\,S2} idf(w)}{\sum_{w \in S2} idf(w)};$$

2.3 Contextual Sequence Matching (Modified BLEU score)

We used a modified version of Bleu score to measure n-gram sequences matching, where for sentence pair s1, s2 we align the matched words between them (through exact, stem, synonyms match respectively). Bleu score as presented by (K. Papineni et al., 2002) is an automated method for evaluating Machine Translation. It compares *n*-grams of the candidate translation with the *n*-grams of the reference human translation and counts the number of matches. These matches are position independent, where candidate translations with unmatched length to reference translations are penalized with *Sentence brevity penalty*.
This helps in measuring n-gram similarity in sentences structure. We define "matched sequence" of a sentence S1 as the sequence of words {wi, wi+1, wi+2, ….. wj}, where wi, and wj are the first and last words in sentence S1 that are matched with words in S2.
For example in sentence pair S1, S2:
S1: Today's great Pax Europa and today's pan-European prosperity depend on this.
S2: Large Pax Europa of today, just like current prosperity paneuropéenne, depends on it.
After stemming:
S1: todai's great pax europa and todai's pan-european prosper depend on thi.
S2: larg pax europa of todai, just like current prosper paneuropéenn, depend on it.

"Matched sequence of S1":
[**todai** 's great **pax** **europa** todai 's pan - european **prosper** **depend**]
"Matched sequence of S2":
[**pax** **europa** **todai** just like current **prosper** paneuropéenn **depend**]

We measure the Bleu score such that:
Bleu{S1, S2} = &BLEU(S1_stemmed,"*Matched sequence of S2*");
Bleu{S2, S1} = &BLEU(S2_stemmed,"*Matched sequence of S1*");
The objective of trimming the excess words outside the "Matched Sequence" range, before matching is to make use of the *Sentence brevity penalty* in case sentence pair S1, S2 may be not similar but having matched lengths.

2.4 Named Entities Matching

Named entities carry an important portion of sentence semantics. For example:
Sentence1: In **Nigeria** , **Chevron** has been accused by the **All - Ijaw** indigenous **people** of **instigating violence** against them and actually paying **Nigerian soldiers** to shoot **protesters** at the **Warri naval base** .
Sentence2: In **Nigeria** , the whole ijaw indigenous showed **Chevron** to encourage the **violence** against them and of up to pay **Nigerian soldiers** to shoot the **demonstrators** at the **naval base** from **Warri** .
The underlined words are Named entities of different types "COUNTRY, ORG, PEOPLE, LOC, EVENT_VIOLENCE" which capture the most important information in each sentence. Thus named entities matching is a measure of semantic matching between the sentence pair.

3 System Description

3.1 Data Resources and Processing

All data is tokenized, stemmed, and stop words are cleaned.

Corpus based resources:

i. **Inverse Document Frequency (IDF) language resource:** The document frequency ***df(t)*** of a term ***t*** is defined as the number of documents in a large collection of documents that contain a term ***"t"***. Terms that are likely to appear in most of the corpus documents reflect less importance than words that appear in specific documents only. That's why the **Inverse Document Frequency** is used as a measure of term importance in information retrieval and text mining tasks. We used the LDC English Gigaword Fifth Edition (LDC2011T07) to generate our idf dictionary. LDC Gigaword contains a huge collection of newswire from (afp, apw, cna, ltw, nyt, wpb, and xin). The generated idf resource contains 5,043,905 unique lower cased entries, and then we generated a stemmed version of the idf dictionary contains 4,677,125 entries. The equation below represents the idf of term t where N is the total number of documents in the corpus.

$$idf_t = \log \frac{N}{df_t}$$

ii. **English Synonyms Dictionary:** Using the Phrase table of an Arabic-to-English Direct Translation Model, we generated English-to-English phrase table using the double-link of English-to-Arabic and Arabic-to-English phrase translation probabilities over all pivot Arabic phrases. Then English-to-English translation probabilities are normalized over all generated English synonyms. **(**Chris Callison-Burch et al, 2006**)** used a similar technique to generate paraphrases to improve their SMT system**.** Figure (1) shows the steps:

```
For each English Phrase "e1"
{
  @ar_phrases = list of Arabic Phrases aligned to "e"
  in the phrase table;
  For each a (@ar_phrases)
   {
        @en_phrases = list of English phrases aligned
        to "a" in the phrase table;

         For each e2 (@en_phrases)
         {
            $Prob(e2\e1) = Prob(a\e1)*Prob(e2\a);
         }
   }
}
```

Figure(1) English phrase-to-phrase synonyms generation from E2A phrase table.

In our system we used the phrase table of the Direct Translation Model 2 (DTM2) (Ittycheriah and Roukos, 2007) SMT system, where each sentence pair in the training corpus was word-aligned, e.g. using a MaxEnt aligner (Ittycheriah and Roukos, 2005) or an HMM aligner (Ge, 2004). then Block Extraction step is done. The generated phrase table contains candidate phrase to phrase translation pairs with source-to-target and target-to source translation probabilities. However the open source Moses SMT system (Koehn et al., 2007)

can be used in the same way to generate a synonyms dictionary from phrase table.

By applying the steps in figure (1):
a) English phrase-to-phrase synonyms table (or English-to-English phrase table), by applying the steps in a generic way.
b) English word-to-word synonyms table, by limiting the generation over English single word phrases.
For example, to get all possible synonyms of the English word "bike", we used all the Arabic phrases that are aligned to "bike" in the phrase table {دراجة, الدراجات , البسكليت, البسكلات},
P: 1905645 14 0.0142582 0.170507 | دراجة | bike |
P: 1910841 25 0.0262152 0.221198 | الدراجات | bike |
P: 2127826 4 0.0818182 0.0414747 | البسكليت | bike |
P: 2396796 2 0.375 0.0138249 | البسكلات | bike |
then we get all the English words in the phrase table aligned to these Arabic translations {دراجة, الدراجات , البسكليت, البسكلات}
This results in an English word-to-word synonyms list for the word "bike" like this:

bike:
motorcycle 0.365253185010659
bicycle 0.198195663512781
cycling 0.143290354808692
motorcycles 0.0871686646772204
bicycles 0.0480779974950311
cyclists 0.0317670845504069
motorcyclists 0.0304152910853553
cyclist 0.0278451740161998
riding 0.0215366691148431
motorbikes 0.0148697281155676

Dictionary based resources:

- **WordNet (Miller, 1995):** is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. WordNet groups words together based on their meanings and interlinks not just word forms—strings of letters—but specific senses of words. As a result, words that are found in close proximity to one another in the network are semantically disambiguated. Second, WordNet labels the semantic relations among words. Using WordNet, we can measure the semantic similarity or relatedness between a pair of concepts (or word senses), and by extension, between a pair of sentences. We use the similarity measure described in (Wu and Palmer, 1994) which finds the path length to the root node from the least common subsumer (LCS) of the two word senses which is the most specific word sense they share as an ancestor.

3.2 Feature Combination

The feature combination step uses the pre-computed similarity scores. Each of the text similarity features can be given a weight that sets its importance. Mathematically, the text similarity score between two sentences can be formulated using a cost function weighting the similarity features as follows: N.B.: The similarity score according to the features above is considered as a directional score.

Similarity(s1, s2) = [w1*Lexical_Score(s1, s2) +
w2*IDF_Lexical_Score(s1, s2) +
w3*Modified_BLEU(s1, s2) +
w4*NE_Score(s1, s2)] / (w1+w2+w3+w4)
Similarity(s2, s1) = [w1*Lexical_Score(s2, s1) +
w2*IDF_Lexical_Score(s2, s1) +
w3*Modified_BLEU(s2, s1) +
w4*NE_Score(s2, s1)] / (w1+w2+w3+w4)
Overall_Score = 5/2*[Similarity(s1, s2)+Similarity(s2, s1)]

where w1, w2, w3, w4 are the weights assigned to the similarity features (lexical, idf-weighted, modified_BLEU, and NE_Match features respectively). The similarity score will be normalized over (w1+w2+w3+w4).
In our experiments, the weights are tuned manually without applying machine learning techniques. We used both *SEM 2012 training and testing data sets for tuning these weights to get the best feature weighting combination to get highest Pearson Correlation score.

4 Experiments and Results

Submitted Runs
Our experiments showed that some features are more dominant in affecting the similarity scoring than others. We performed a separate experiment for each of the four feature types to illustrate their effect on textual semantic similarity measurement

using direct lexical matching, stemming matching, synonyms matching, as well as (stem+synonyms) matching. Table (1) reports the mean Pearson correlation results of these experiments on STS2012-test dataset

	Direct	**Stem only**	**Synonyms only**	**Synonyms + Stem**
NE	0.303	0.297	0.306	0.304
BLEU	0.439	0.446	0.469	0.453
Lexical	0.59	0.622	0.611	0.624
IDF	0.488	0.632	0.504	**0.634**

Table (1) reports the mean Pearson score for NE, BLEU, Lexical, and idf-weighted matching features respectively on STS2012-test dataset.

The submitted runs IBM_EG-run2, IBM_EG-run5, IBM_EG-run6 are the three runs with feature weighting and experiment set up that performed best on STS 2012 training and testing data sets.

Run 2: In this run the word matching was done on exact, and synonyms match only. Stemmed word matching was not introduced in this experiment. we tried the following weighting between similarity feature scores, where we decreased the weight of BLEU scoring feature to 0.5, and increased the idf_Lexical match weight of 3.5. this is because our initial tuning experiments showed that increasing the idf lexical weight compared to BLEU weight gives improved results. The NE matching feature weight was as follows:

NE_weight = 1.5* percent of NE word to sentence word count
= 1.5* (NE_words_count/Sentence_word_count)

Run 5: In this experiment we introduced Porter stemming word matching, as well as stemmed synonyms matching (after generating a stemmed version of the synonyms dictionary). BLEU score feature was removed from this experiment, while keeping the idf-weight= 3, lexical-weight = 1, and NE-matching feature weight = 1.

Run 6: For this run we kept only IDF-weighted lexical matching feature which proved to be the dominant feature in the previous runs, in addition to Porter stemming word matching, and stemmed synonyms matching.

Data: the training data of STS 2013 Core task consist of the STS 2012 train and test data. This data covers 5 datasets: paraphrase sentence pairs (MSRpar), sentence pairs from video descriptions (MSRvid), MT evaluation sentence pairs (SMTnews and SMTeuroparl) and gloss pairs (OnWN).

Results on Training Data

System outputs will be evaluated according to the official scorer which computes weighted Mean Pearson Correlation across the evaluation datasets, where the weight depends on the number of pairs in each dataset.

Table (2), reports the results achieved on each of the STS 2012 training dataset. While table (3), reports the results achieved on STS 2012 test dataset.

	IBM_run2	**IBM_run5**	**IBM_run6**
Mean	0.59802	0.64170	**0.68395**
MSRpar	0.61607	**0.63870**	0.62629
MSRvid	0.70356	0.80879	**0.83722**
SMTeuroparl	0.47173	0.47403	**0.58627**

Table (2) Results on STS 2012 training datasets.

	IBM_run2	**IBM_run5**	**IBM_run6**
Mean	0.59408	0.62614	**0.63365**
MSRpar	0.56059	0.59108	**0.61306**
MSRvid	0.73189	0.79960	**0.87154**
SMTeuroparl	0.51480	**0.50563**	0.41298
OnWN	0.62927	0.65760	**0.67136**
SMTnews	0.42305	**0.44551**	0.40819

Table (3) Results on STS 2012 test datasets.

Results on Test Data:

The best configuration of our system was **IBM_EG-run6** which was ranked #11 for the evaluation metric Mean (r = 0.5502) when submitted during the task evaluation period . **Run6** as illustrated before was planned to measure idf-weighted lexical matching feature only, over Porter stemmed, and stemmed synonyms words. **However** when revising this experiment set up

during preparing the paper, after the evaluation period, we found that the English-to-English synonyms table was not correctly loaded during matching, thus skipping synonyms matching feature from this run. So the official result **IBM_EG-run6** reports only idf-weighted matching over Porter stemmed bag of words. By fixing this and replicating the experiment **IBM_EG-run6-UnOfficial** as planned to be**,** the mean Pearson correlation jumps 4 points (**r = 0.59221**) which ranks this system as the 3rd system among 90 submitted systems very slightly below the 2nd system (only 0.0006 difference on the mean correlation metric). In table (4), we report the official results achieved on STS 2013 test data. While table (5), reports the unofficial results achieved after activating the synonyms matching feature in **IBM_EG-run6 (unofficial)** and comparing this run to the best two reported systems.

	IBM_EG-run2	**IBM_EG-run5**	**IBM_EG-run6**
headlines	0.7217	0.7410	**0.7447**
OnWN	0.6110	0.5987	**0.6257**
FNWN	0.3364	0.4133	**0.4381**
SMT	**0.3460**	0.3426	0.3275
Mean	0.5365	0.5452	**0.5502**
Rank	**#19**	**#15**	**#11**

Table (4) Official Results on STS 2013 test datasets.

	UMBC_EBIQUITY-ParingWords	UMBC_EBIQUITY-galactus	**IBM_EG-run6 (UnOfficial)**
headlines	0.7642	0.7428	**0.77241**
OnWN	0.7529	0.7053	**0.70103**
FNWN	0.5818	0.5444	**0.44356**
SMT	0.3804	0.3705	**0.36807**
Mean	0.6181	0.5927	**0.59221**
Rank	#1	#2	**#3**

Table (5) UnOfficial Result after activating the synonyms matching feature in **IBM_EG-run6** compared to the best two performing systems in the evaluation.

Results of un-official run:

One unofficial run was performed after the evaluation submission deadline due to the tight schedule of the evaluation. This experiment introduces the effect of WordNet Wu and Palmer similarity measure on the configuration of Run5 (Porter stemming word matching, with synonyms matching, zero weight for BLEU score feature, while keeping the idf-weight= 3, lexical-weight = 1, and NE-matching feature weight = 1)

Table (6) reports the unofficial result achieved on STS 2013 test data, compared to the Official run **IBM_Eg-run5**.

	Unofficial-Run	**IBM_EG-run5**
Mean	0.52682	**0.5452**
headlines	0.70018	**0.7410**
OnWN	0.60371	0.5987
FNWN	0.35691	**0.4133**
SMT	0.33875	**0.3426**

Table (6) Un-Official Result on STS 2013 test datasets.

From the results in Table (6) it is clear that Corpus based synonyms matching outperforms dictionary-based WordNet matching over SEM2013 testset.

5 Conclusion

We proposed an unsupervised approach for measuring semantic textual similarity based on Lexical matching features (with porter stemming matching and synonyms matching), idf-Lexical matching features, Ngram Frquency (Modified BLEU) matching feature, as well as Named Entities matching feature combined together with a weighted cost function. Our experiments proved that idf-weighted Lexical matching in addition to porter stemming and synonyms-matching features perform best on most released evaluation datasets. Our best system officially ranked number 11 among 90 participating system reporting a Pearson Mean correlation score of 0.5502. However our best experimental set up "idf-weighted Lexical matching in addition to porter stemming and synonyms-matching" reported in an unofficial run a mean correlation score of **0.59221** which ranks the system as number 3 among the 90 participating systems. In our future work we intend to try some machine learning algorithms (like AdaBoost for

example) for weighting our similarity matching feature scores. Also we plan to extend the usage of synonyms matching from the word level to the n-gram phrase matching level, by modifying the BLEU Score N-gram matching function to handle synonym phrases matching.

Acknowledgments

We would like to thank the reviewers for their constructive criticism and helpful comments.

References

Alfred. V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

American Psychological Association. 1983. *Publications Manual.* American Psychological Association, Washington, DC.

Association for Computing Machinery. 1983. *Computing Reviews*, 24(11):503-512.

Ashok K. Chandra, Dexter C. Kozen, and Larry J.Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114-133.

C. Y. Lin and E. H. Hovy. 2003. *Automatic evaluation of summaries using n-gram co-occurrence statistics.* In Proceedings of Human Language Technology Conference (HLT-NAACL 2003), Edmonton, Canada, May.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. *Improved statistical machine translation using paraphrases.* In Proceedings of HLT-NAACL.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

G. Salton and C. Buckley. 1997. *Term weighting approaches in automatic text retrieval.* In Readings in Information Retrieval. Morgan Kaufmann Publishers, San Francisco, CA.

Ittycheriah, A. and Roukos, S. (2007). *Direct translation model 2*. In Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, pp.57–64, Rochester, NY.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. *Bleu: a method for automatic evaluation of machine translation.* In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Cambridge, UK.

M. Lapata and R. Barzilay. 2005. *Automatic evaluation of text coherence: Models and representations*. In Proceedings of the 19th International Joint Conference on Artificial Intelligence, Edinburgh.

P. Koehn, F.J. Och, and D. Marcu. 2003. *Statistical Phrase-Based Translation*. Proc. Of the Human Language Technology Conference, HLTNAACL' 2003, May.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondˇrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. *Moses: Open Source Toolkit for Statistical Machine Translation.* In Proceedings of the ACL 2007 Demo and Poster Sessions, pages 177–180.

R. Mihalcea , C. Corley, and C. Strapparava 2006. *Corpus-based and knowledge-based measures of text semantic similarity.* In Proceedings of the American Association for Artificial Intelligence. (Boston, MA).

Satanjeev Banerjee and Ted Pedersen. 2003. *Extended gloss overlaps as a measure of semantic relatedness*. In Proceedings of the 18th International Joint Conference on Artificial Intelligence, pages 805–810.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi, 2004, *WordNet::Similarity - Measuring the Relatedness of Concepts*. Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2004).

Wu, Z., and Palmer, M. 1994. *Verb semantics and lexical selection.* In 32nd Annual Meeting of the Association for Computational Linguistics, 133–138.

Weiwei Guo and Mona Diab. 2012a. *Learning the latent semantics of a concept from its definition*. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics.