JU_CSE_NLP: Multi-grade Classification of Semantic Similarity Between Text Pairs

Snehasis Neogi[1], Partha Pakray[2], Sivaji Bandyopadhyay[1]
[1]Computer Science & Engineering Department
Jadavpur University, Kolkata, India
[2]Computer Science & Engineering Department
Jadavpur University, Kolkata, India
Intern at Xerox Research Centre Europe
Grenoble, France
{snehasis1981,parthapakray}@gmail.com
sbandyopadhyay@cse.jdvu.ac.in

Alexander Gelbukh
Center for Computing Research
National Polytechnic Institute
Mexico City, Mexico
gelbukh@gelbukh.com

Abstract

This article presents the experiments carried out at Jadavpur University as part of the participation in Semantic Textual Similarity (STS) of Task 6 @ Semantic Evaluation Exercises (SemEval-2012). Task-6 of SemEval- 2012 focused on semantic relations of text pair. Task-6 provides five different text pair files to compare different semantic relations and judge these relations through a similarity and confidence score. Similarity score is one kind of multi way classification in the form of grade between 0 to 5. We have submitted one run for the STS task. Our system has two basic modules - one deals with lexical relations and another deals with dependency based syntactic relations of the text pair. Similarity score given to a pair is the average of the scores of the above-mentioned modules. The scores from each module are identified using rule based techniques. The Pearson Correlation of our system in the task is 0.3880.

1 Introduction

Task-6[1] [1] of SemEval-2012 deals with semantic similarity of text pairs. The task is to find the similarity between the sentences in the text pair (s1 and s2) and return a similarity score and an optional confidence score. There are five datasets in the test data and with tab separated text pairs. The datasets are as follows:

- MSR-Paraphrase, Microsoft Research Paraphrase Corpus (750 pairs of sentences.)
- MSR-Video, Microsoft Research Video Description Corpus (750 pairs of sentences.)
- SMTeuroparl: WMT2008 development dataset (Europarl section) (459 pairs of sentences.)
- SMTnews: news conversation sentence pairs from WMT.(399 pairs of sentences.)
- OnWN: pairs of sentences where the first comes from Ontonotes and the second from a WordNet definition. (750 pairs of sentences.)

Similarity score ranges from 0 to 5 and confidence score from 0 to 100. An s1-s2 pair gets a similarity score of 5 if they are completely equivalent. Similarity score 4 is allocated for mostly equivalent s1-s2 pair. Similarly, score 3 is allocated for roughly equivalent pair. Score 2, 1 and 0 are allocated for non-equivalent details sharing, non-equivalent topic sharing and totally different pairs respectively. Major challenge of this task is to find the similarity score based similarity for the text pair. Generally text entailment tasks refer whether sentence pairs are entailed or not: binary classification (YES, NO) [2] or multi-classification (Forward, Backward, bidirectional or no entailment) [3][4]. But multi grade classification of semantic similarity assigns a score to the sentence pair. Our system considers lexical and dependency based syntactic measures for semantic similarity. Similarity scores are the basic average of these module scores. A subsequent

[1] http://www.cs.york.ac.uk/semeval-2012/task6/

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 571–574,
Montréal, Canada, June 7-8, 2012. ©2012 Association for Computational Linguistics

section describes the system architecture. Section 2 describes JU_NLP_CSE system for STS task. Section 3 describes evaluation and experimental results. Conclusions are drawn in Section 4.

2 System Architecture

The system of Semantic textual similarity task has two main modules: one is lexical module and another one is dependency parsing based syntactic module. Both these module have some preprocessing tasks such as stop word removal, co-reference resolution and dependency parsing etc. Figure 1 displays the architecture of the system.

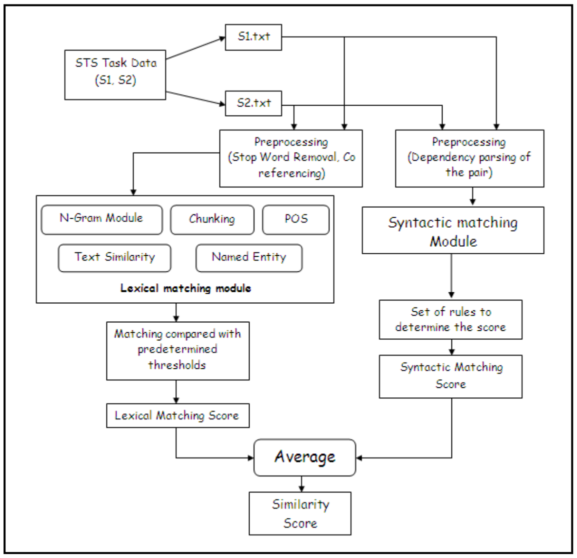

Figure 1: System Architecture

2.1 Pre-processing Module

The system separates the s1-s2 sentence pairs contained in the different STS task datasets. These separated pairs are then passed through the following sub modules:

i. Stop word Removal: Stop words are removed from s1 - s2 sentence pairs.

ii. Co-reference: Co-reference resolutions are carried out on the datasets before passing through the TE module. The objective is to increase the score of the entailment percentage. A word or phrase in the sentence is used to refer to an entity introduced earlier or later in the discourse and both having same things then they have the same referent or co reference. When the reader must look back to the previous context, reference is called "*Anaphoric Reference*". When the reader must look forward, it is termed "*Cataphoric Reference*". To address this problem we used a tool called JavaRAP[2] (A java based implementation of Anaphora Procedure (RAP) - an algorithm by Lappin and Leass (1994)).

iii. Dependency Parsing: Separated s1 – s2 sentences are parsed using Stanford dependency parser[3] to produce the dependency relations in the texts. These dependency relations are used for WordNet based syntactic matching.

2.2 Lexical Matching Module

In this module the TE system calculates different matching scores such as N – Gram match, Text Similarity, Chunk match, Named Entity match and POS match.

i. N-Gram Match module: The N-Gram match basically measures the percentage match of the unigram, bigram and trigram of hypothesis present in the corresponding text. These scores are simply combined to get an overall N – Gram matching score for a particular pair.

ii. Chunk Match module: In this sub module our system evaluates the key NP-chunks of both text (s1) and hypothesis (s2) using NP Chunker v1.1[3] (The University of Sheffield). The hypothesis NP chunks are matched in the text NP chunks. System calculates an overall value for the chunk matching, i.e., number of text NP chunks that match the hypothesis NP chunks. If the chunks are not similar in their surface form then our system goes for wordnet synonyms matching for the words and if they match in wordnet synsets information, it will be encountered as a similar chunk. WordNet [5] is one of most important resource for lexical analysis. The WordNet 2.0 has been used for WordNet based chunk matching. The API for WordNet Searching (JAWS)[4] is an API that provides Java applications with the ability to retrieve data from the WordNet synsets.

iii. Text Similarity Module: System takes into consideration several text similarities calculated

[2] http://aye.comp.nus.edu.sg/~qiu/NLPTools/JavaRAP.html
[3] http://www.dcs.shef.ac.uk/~mark/phd/software/
[4] http://lyle.smu.edu/~tspell/jaws/index.html

over the s1-s2 pair. These text similarity values are summed up to produce a total score for a particular s1-s2 pair. Major Text similarity measures that our system considers are:

- ➢ *Cosine Similarity*
- ➢ *Lavenstine Distance*
- ➢ *Euclidean Distance*
- ➢ *MongeElkan Distance*
- ➢ *NeedlemanWunch Distance*
- ➢ *SmithWaterman Distance*
- ➢ *Block Distance*
- ➢ *Jaro Similarity*
- ➢ *MatchingCoefficient Distance*
- ➢ *Dice Similarity*
- ➢ *OverlapCoefficient*
- ➢ *QGrams Distance*

iv. Named Entity Matching: It is based on the detection and matching of Named Entities in the s1-s2 pair. Stanford Named Entity Recognizer[5] is used to tag the named entities in both s1 and s2. System simply maps the number of hypothesis (s2) NEs present in the text (s1). A score is allocated for the matching.

NE_match = (Number of common NEs in Text and Hypothesis) / (Number of NE in Hypothesis).

v. Part –of – Speech (POS) Matching: This module basically deals with matching the common POS tags between s1 and s2 sentences. Stanford POS tagger[6] is used to tag the part of speech in both s1 and s2. System matches the verb and noun POS words in the hypothesis that match in the text. A score is allocated based on the number of POS matching.

POS_match = (Number of common verb and noun POS in Text and Hypothesis) / (Total number of verb and noun POS in hypothesis).

System calculates the sum of the entire sub module (modules described in section 2.2) scores and forms a single percentage score for the lexical matching. This score is then compared with some predetermined threshold value to assign a final lexical score for each pair. If percentage value is above 0.80 then lexical score 5 is allocated. If the value is between 0.60 to 0.80 then lexical score 4 is allocated. Similarly, lexical score 3 is allocated for percentage score of 0.40 to 0.60 and so on. One lexical score is finally generated for each text pair.

2.3. Syntactic Matching Module:

TE system considers the preprocessed dependency parsed text pairs (s1 – s2) and goes for word net based matching technique. After parsing the sentences, they have some attributes like subject, object, verb, auxiliaries and prepositions tagged by the dependency parser tag set. System uses these attributes for the matching procedure and depending on the nature of matching a score is allocated to the s1-s2 pair. Matching procedure is basically done through comparison of the following features that are present in both the text and the hypothesis.

- • *Subject – Subject comparison.*
- • *Verb – Verb Comparison.*
- • *Subject – Verbs Comparison.*
- • *Object – Object Comparison.*
- • *Cross Subject – Object Comparison.*
- • *Object – Verbs Comparison.*
- • *Prepositional phrase comparison.*

Each of these comparisons produces one matching score for the s1-s2 pair that are finally combined with previously generated lexical score to generate the final similarity score by taking simple average of lexical and syntactic matching scores. The basic heuristics are as follows:

(i) If the feature of the text (s1) directly matches the same feature of the hypothesis (s2), matching score 5 is allocated for the text pair.

(ii) If the feature of either text (s1) or hypothesis (s2) matches with the wordnet synsets of the corresponding text (s1) or hypothesis (s2), matching score 4 is allocated.

(iii) If wordnet synsets of the feature of the text (s1) match with one of the synsets of the feature of the hypothesis (s2), matching score 3 is given to the pair.

(iv) If wordnet synsets of the feature of either text (s1) or hypothesis (s2) match with the synsets of the corresponding text (s1) or hypothesis (s2) then matching score 2 is allocated for the pair.

[5] http://nlp.stanford.edu/software/CRF-NER.shtml

[6] http://nlp.stanford.edu/software/tagger.shtml

(v) Similarly if in both the cases match occurs in the second level of wordnet synsets, matching score 1is allocated.
(vi) Matching score 0 is allocated for the pair having no match in their features.
After execution of the module, system generates some scores. Lexical module generates one lexical score and wordnet based syntactic matching module generates seven matching scores. At the final stage of the system all these scores are combined and the mean is evaluated on this combined score. This mean gives the similarity score for a particular s1-s2 pair of different datasets of STS task. Optional confidence score is also allocated which is basically the similarity score multiplied by 10, i.e., if the similarity score is 5.22, the confidence score will be 52.2.

3. Experiments on Dataset and Result

We have submitted one run in SemEval-2012 Task 6. The results for Run on STS Test set are shown in Table 1.

task6-JU_CSE_NLP-Semantic_Syntactic_Approach	**Correlations**
ALL	0.3880
ALLnrm	0.6706
Mean	0.4111
MSRpar	0.3427
MSRvid	0.3549
SMT-eur	0.4271
On-WN	0.5298
SMT-news	0.4034

Table 1: Results of Test Set

ALL: Pearson correlation with the gold standard for the five datasets and the corresponding rank 82.
ALLnrm: Pearson correlation after the system outputs for each dataset are fitted to the gold standard using least squares and the corresponding rank 86.
Mean: Weighted mean across the 5 datasets, where the weight depends on the number of pairs in the dataset and the corresponding rank 76.
The subsequent rows show the pearson correlation scores for each of the individual datasets.

4. Conclusion

Our JU_CSE_NLP system for the STS task mainly focus on lexical and syntactic approaches. There are some limitations in the lexical matching module that shows a correlation that is not higher in the range. In case of simple sentences lexical matching is helpful for entailment but for complex and compound sentences the lexical matching module loses its accuracy. Semantic graph matching or conceptual graph implementation can improve the system. That is not considered in our present work. Machine learning tools can be used to learn the system based on the features. It can also improve the correlation. In future work our system will include semantic graph matching and a machine-learning module.

Acknowledgments

The work was done under support of the DST India-CONACYT Mexico project "*Answer Validation through Textual Entailment*" funded by DST, Government of India.

References

[1] Eneko Agirre, Daniel Cer, Mona Diab and Aitor Gonzalez. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), in conjunction with the First Joint Conference on Lexical and Computational Semantics (*SEM 2012). (2012)

[2] Dagan, I., Glickman, O., Magnini, B.: *The PASCAL Recognising Textual Entailment Challenge.* Proceedings of the First PASCAL Recognizing Textual Entailment Workshop. (2005).

[3] H. Shima, H. Kanayama, C.-W. Lee, C.-J. Lin,T. Mitamura, S. S. Y. Miyao, and K. Takeda. *Overview of ntcir-9 rite: Recognizing inference in text.* In NTCIR-9 Proceedings,2011.

[4] Pakray, P., Neogi, S., Bandyopadhyay, S., Gelbukh, A.: *A Textual Entailment System using Web based Machine Translation System*. NTCIR-9, National Center of Sciences, Tokyo, Japan. December 6-9, 2011. (2011).

[5] Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998).