FCICU: The Integration between Sense-Based Kernel and Surface-Based Methods to Measure Semantic Textual Similarity

Basma Hassan
Computer Science Department, Faculty of Computers and Information
Fayoum University
Fayoum, Egypt
bhassan@fayoum.edu.eg

Samir AbdelRahman
Computer Science Department, Faculty of Computers and Information
Cairo University
Giza, Egypt
s.abdelrahman@fci-cu.edu.eg

Reem Bahgat
Computer Science Department, Faculty of Computers and Information
Cairo University
Giza, Egypt
r.bahgat@fci-cu.edu.eg

Abstract

This paper describes FCICU team participation in SemEval 2015 for Semantic Textual Similarity challenge. Our main contribution is to propose a word-sense similarity method using BabelNet relationships. In the English subtask challenge, we submitted three systems (runs) to assess the proposed method. In Run1, we used our proposed method coupled with a string kernel mapping function to calculate the textual similarity. In Run2, we used the method with a tree kernel function. In Run3, we averaged Run1 with a previously proposed surface-based approach as a kind of integration. The three runs are ranked 41st, 57th, and 20th of 73 systems, with mean correlation 0.702, 0.597, and 0.759 respectively. For the interpretable task, we submitted a modified version of Run1 achieving mean F1 0.846, 0.461, 0.722, and 0.44 for alignment, type, score, and score with type respectively.

1 Introduction

Semantic Textual Similarity (STS) is the task of measuring the similarity between two text snippets according to their meaning. Human has an intrinsic ability to recognize the degree of similarity and difference between texts. Simulating the process of human judgment in computers is still an extremely difficult task and has recently drawn much attention. STS is very important because a wide range of NLP applications such as information retrieval, question answering, machine translation, etc. rely heavily on this task.

This paper describes our proposed STS systems by which we participated in two subtasks of STS task (Task2) at SemEval 2015, namely English STS and Interpretable STS. The former calculates a graded similarity score from 0 to 5 between two sentences (with 5 being the most similar), while the latter is a pilot subtask that requires aligning chunks of two sentences, describing what kind of relation exists between each pair of chunks, and a score for the similarity between the pair of chunks (Agirre et al., 2015).

Sense or meaning of natural language text can be inferred from several linguistic concepts, including lexical, syntactic, and semantic knowledge of the language. Our approach employs those aspects to calculate the similarity between senses of text constituents, phrases or words, relying mainly on BabelNet senses. The similarity between two text snippets is firstly calculated using kernel functions, which map a text snippet to the feature space based on a proposed word sense similarity method. Besides, the sense-based similarity score obtained is combined with a surface-based similarity score to study the consolidation impact in the STS task.

The paper is organized as follows. Section 2 explains our proposed word sense similarity method. Section 3 describes the proposed systems. Section 4 presents the experiments conducted and analyzes the results achieved. Section 5 concludes the paper and suggests some future directions.

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 154–158,
Denver, Colorado, June 4-5, 2015.

2 The proposed Word-Sense Similarity (WSS) Method

Several semantic textual similarity (STS) methods have been proposed in literature. Sense-based methods are qualified when different words are used to convey the same meaning in different texts (Pilehvar et al., 2013). Surface-based methods, mostly fail in identifying similarity between texts with maximal semantic overlap but minimal lexical overlap. We present a sense-based STS approach that produces similarity score between texts by means of a kernel function (Shawe-Taylor and Cristianini, 2004). Then, we integrate the sense-based approach with the surface-based soft cardinality approach presented in (Jimenez et al., 2012) to demonstrate that both sense-based and surface-based similarity methods are complementary to each other.

The design of our kernel function relies on the hypothesis that the greater the similarity of word senses between two texts, the higher their semantic equivalence will be. Accordingly, our kernel maps a text to feature space using a similarity measure between word senses. We proposed a WSS measure that computes the similarity score between two word senses (ws_i, ws_j) using the arithmetic mean of two measures: *Semantic Distance* (sim_D) and *Contextual Similarity* (sim_C). That is:

$$WSS(ws_i, ws_j) = \frac{sim_D(ws_i, ws_j) + sim_C(ws_i, ws_j)}{2} \quad (1)$$

2.1 Semantic Distance

This measure computes the similarity between word senses based on the distance between them in a multilingual semantic network, named BabelNet (Navigli and Ponzetto, 2010). BabelNet[1] is a rich semantic knowledge resource that covers a wide range of concepts and named entities connected with large numbers of semantic relations. Concepts and relations are gathered from *WordNet* (Miller, 1995); and *Wikipedia*[2]. The semantic knowledge is encoded as a labeled directed graph, where vertices are BabelNet senses (concepts), and edges connect pairs of senses with a label indicating the type of the semantic relation between them. Our semantic distance measure is a function of two similarity scores: $\boldsymbol{sim_{Bn}}$ and $\boldsymbol{sim_{NBn}}$.

The first score (sim_{Bn}) is based on the distance between two word-senses, ws_i and ws_j; where, the shorter the distance between them, the more semantically related they are. That is:

$$sim_{Bn}(ws_i, ws_j) = 1 - \frac{len(ws_i, ws_j)}{Maxlen} \quad (2)$$

where $Maxlen$[3] is the maximum path length connecting two senses in BabelNet, and $len(ws_i,ws_j)$ is the length of the shortest path between two senses, ws_i and ws_j, in BabelNet in both directions; i.e $ws_i \rightarrow ws_j$, and $ws_j \rightarrow ws_i$. The shortest path is calculated using Dijkstra's algorithm.

The second score (sim_{NBn}) represents the degree of similarity between the neighbors of ws_i and the neighbors of ws_j, which influences the degree of similarity between the two senses. Hence, sim_{NBn} is calculated by taking the arithmetic mean of all neighbor-pairs similarity. That is:

$$sim_{NBn}(ws_i, ws_j) = \frac{1}{n_i \times n_j} \sum_{ws_k \in NS_i} \sum_{ws_l \in NS_j} sim_{WuP}(ws_k, ws_l) \quad (3)$$

where NS_i and NS_j are the sets of the most semantically related senses directly connected to ws_i and ws_j respectively in BabelNet; $n_i = |NS_i|$, and $n_j = |NS_j|$; and $sim_{WuP}(ws_k, ws_l)$ is Wu and Palmer similarity measure (Wu and Palmer, 1994).

The values of the two scores presented above determine the way of calculating the semantic distance measure (sim_D) for word senses' pair (ws_i, ws_j). For zero similarity of both scores, sim_D is simply equals to Wu and Palmer similarity measure; i.e. $sim_D(ws_i,ws_j) = sim_{WuP}(ws_i,ws_j)$. Generally, for non-zero similarity scores, sim_D is calculated using the arithmetic mean of the two scores.

2.2 Contextual Similarity

This measure calculates the similarity between the word senses pair (ws_i, ws_j) based on the overlap between their contexts derived from a corpus. The overlap coefficient used is *Jaccard Coefficient*. That is:

$$sim_C(ws_i, ws_j) = \frac{|C_i \cap C_j|}{|C_i \cup C_j|} \quad (4)$$

where C_i is the set of: 1) all the word senses that co-occur with ws_i in the corpus, and 2) all senses directly connected to ws_i in BabelNet; C_j is similar.

[1] http://babelnet.org/

[2] http://en.wikipedia.org/

[3] We tried different values in experiments and the best was 7.

3 Systems Description

3.1 Text Preprocessing

The given input sentences are first preprocessed to map the raw natural language text into structured or annotated representation. This process includes different tasks: tokenization, lemmatization, Part-of-Speech tagging, and word-sense tagging. All tasks except word-sense tagging are carried out using Stanford CoreNLP (Manning et al., 2014). Sense tagging is the task of attaching a sense to a word or a token. It is performed by selecting the most commonly used BabelNet sense that matches the part of speech (POS) of the word. Accordingly, we restricted sense tagging to: nouns, verbs, adjectives, and adverbs.

3.2 English STS Subtask

We submitted three systems in this subtask, named Run1, Run2, and Run3.

3.2.1 Sense-based String Kernel (Run1)

Given two sentences, s_1 and s_2, the similarity score between s_1 and s_2 resulted by this system is the value of a designed string kernel function between the two sentences. This kernel is defined by an embedded mapping from the space of sentences possibly to a vector space F, whose coordinates are indexed by a set I of word senses contained in s_1 and s_2; i.e. $\phi : s \rightarrow (\phi_{ws}(s))_{ws \in I} \in F$. Thus, given a sentence s, it can be represented by a row vector as: ϕ (s) = ($\phi_{ws1}(s)$, $\phi_{ws2}(s)$ … $\phi_{wsN}(s)$), in which each entry records how similar a particular word sense ($ws \in I$) is to the sentence s. The mapping is given by:

$$\phi_{ws}(s) = \max_{1 \le i \le n} \{ WSS(ws, ws_i) \}, \quad (5)$$

where $WSS(ws, ws_i)$ is our defined word sense similarity method (Eq. (1)), and n is the number of word senses contained in sentence s.

The string kernel between two sentences s_1 and s_2 is calculated as (Shawe-Taylor and Cristianini, 2004):

$$\kappa_S(s_1, s_2) = \langle \phi(s_1), \phi(s_2) \rangle = \sum_{ws \in I} \phi_{ws}(s_1) \cdot \phi_{ws}(s_2) \quad (6)$$

The last step remaining is normalizing the kernel (i.e. range = [0,1]) to avoid any biasness to sentence length. The normalized string kernel $\kappa_{NS}(s_1,s_2)$ is calculated by (Shawe-Taylor and Cristianini, 2004):

$$\kappa_{NS}(s_1, s_2) = \frac{\kappa_S(s_1, s_2)}{\sqrt{\kappa_S(s_1, s_1)\kappa_S(s_2, s_2)}} \quad (7)$$

Hence, $sim_{Run1}(s_1, s_2) = \kappa_{NS}(s_1, s_2)$.

3.2.2 Sense-based Tree Kernel (Run2)

This system applies tree kernel instead of string kernel. Tree kernels generally map a tree to the feature space of subtrees. There are various types of tree kernel designed in literature, among them is the *all-subtree kernel* presented in (Shawe-Taylor and Cristianini, 2004). The all-subtree kernel is defined by an embedded mapping from the space of all finite syntactic trees to a vector space F, whose coordinates are indexed by a subset T of syntactic subtrees; i.e. $\phi : t \rightarrow (\phi_{st}(t))_{st \in T} \in F$. The mapping $\phi_{st}(t)$ is a simple exact matching function that returns 1 if st is a subtree in t, and returns 0 otherwise. We modified the mapping of all-subtree kernel to capture the semantic similarity between subtrees instead of the structural similarity. The semantic similarity between subtrees is calculated recursively bottom-up from leaves to the root, in which the similarity between leaves is calculated using our defined word sense similarity method.

From this point, the remaining steps are typical to the string kernel steps followed in the first system. Hence, given two sentences s_1 and s_2, their similarity score is the normalized kernel value between their syntactic parse trees t_1 and t_2; i.e. $sim_{Run2}(s_1, s_2) = \kappa_{NT}(t_1, t_2)$.

3.2.3 Sense-based with Surface-based (Run3)

This system provides the results of taking the arithmetic mean of: 1) our sense-based string kernel (Run1); and 2) the surface-based similarity function proposed by Jimenez et al. (2012). The approach presented in (Jimenez et al., 2012) represents sentence words as sets of *q-grams* on which the notion of Soft Cardinality is applied. In this system, all the calculations in the approach are used unchanged with the following parameters set-up: p=2, bias=0, and α=0.5. Accordingly, the similarity function is the Dice overlap coefficient on q-grams; i.e. $sim_{SC}(A, B) = 2|A \cap B|' / (|A|' + |B|')$.

Hence,

$$sim_{Run3}(s_1, s_2) = \frac{(\kappa_{NS}(s_1, s_2) + sim_{SC}(s_1, s_2))}{2} \quad (8)$$

3.3 Interpretable STS Subtask

The interpretable STS is a pilot subtask, which aims to determine the parts of sentences, chunks, that are equivalent in meaning and the parts that are not. This is twofold: (a) aligning corresponding chunks, and (b) assigning a *similarity score*, and a *type* to each alignment. Given two sentences splitted into gold standard chunks, our system carries out the task requirements using our sense-based string kernel by considering each chunk as a text snippet. Firstly, the similarity between chunks of all possible chunk-pairs is calculated, upon which chunks are aligned. Where, chunk pairs with a high similarity score are aligned first, followed by pairs with lower similarity. Thereafter, for each alignment of chunks c_1 and c_2, the alignment type is determined according to the following rules:

- If the similarity score between c_1 and c_2 is 5, the type is EQUI.
- If all word senses of c_1 matched the word senses in c_2, the type is SPEC2; similarly for SPEC1.
- If both c_1 and c_2 contain a single word sense, and are directly connected by an antonym relation in BabelNet, then the type is OPPO.
- If the similarity score between c_1 and c_2 is in range [3,5[, the type is SIM; while if it is in range]0,3[, the type is REL.
- If any chunk has no corresponding chunk in the other sentence, then the type is either NOALI or ALIC based on the alignment restriction in the subtask.

4 Experimental Results

4.1 English STS

The main evaluation measure selected by the task organizers was the mean Pearson correlation between the system scores and the gold standard scores calculated on the test set (3000 sentence pairs from five datasets). Table 1 presents the official results of our submissions in this subtask on SemEval-2015 test set. It also includes the results of the Soft Cardinality STS approach (SC) on the same test set for analysis. Our best system (Run3) achieved 0.7595 and ranked the 20th out of 73 systems.

We conducted preliminary experiments on the training dataset of SemEval-2015 for evaluating our sense-based string and tree kernel similarity methods, and the integration between each of them with the SC approach. The results of those experiments led to the final submission of the two kernels separately (Run1 and Run2) and integrating the string kernel method with SC (Run3). Table 2 focuses on the results obtained from our integrated system (Run3) and SC approach in training, but includes also the recent SC approach (SC-ML) proposed in (Jimenez et al., 2014).

It is noteworthy from the tables that Run3 improved the SC system results on both the training and testing sets for all the different settings for alpha value in the SC approach. The possible reason based on our observation on the training datasets is that the two systems have opposite strength and weakness points. Figure 1 depicts the similarity scores resulted from Run1, Run3, and SC systems along with the gold standard scores (GS) on some sentence pairs from images dataset. It is shown from the figure that Run1 outperforms SC for semantically equivalent sentence pairs (i.e. scores > 3.5), while SC outperforms Run1 for less-related sentence pairs (i.e. score < 2). Hence, their integration by taking their average (Run3) improves the performance of their individual use and did not reduce the SC results. Also, though this integration is simple, it outperformed SC-ML that applies machine learning on some extracted text features.

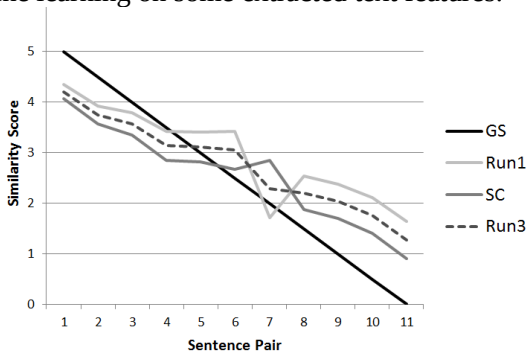

Figure 1. Sample Results of Run1, Run3, and SC on 'images' Dataset of SemEval Training data.

4.2 Interpretable STS

There were two datasets only in the test set, namely images and headlines. The results in this subtask are evaluated by four F1 measures for alignment, score, alignment type, and both score with alignment. The results of our submitted run (average of the two datasets) were 0.846, 0.461, 0.722, and 0.44 for F1-Ali, F1-type, F1-score, and F1-score+type respectively.

System	answers-forums	answers-students	belief	headlines	images	Mean	Rank
Run1	0.6152	0.6686	0.6109	0.7418	0.7853	0.7022	41st/73
Run2	0.3659	0.6460	0.5896	0.6448	0.6194	0.5970	57th/73
Run3	0.7091	0.7096	0.7184	0.7922	0.8223	**0.7595**	20th/73
SC	0.7078	0.7020	0.7232	0.7966	0.8120	0.7565	-

Table 1. Our Results on SemEval-2015 Test Datasets.

α	System	deft-forum	deft-news	headlines	images	OnWN	tweet-news	Mean
-	**Run1**	0.4259	0.7271	0.6914	0.7576	0.7597	0.7227	0.6955
-	**SC-ML**	0.4607	0.7216	0.7605	0.7782	0.8426	0.6583	0.7209
0.25	**Run3**	0.5092	0.7479	0.7383	0.7902	0.7857	0.7744	**0.7387**
	SC	0.5047	0.7311	0.7362	0.7785	0.7727	0.7709	0.7307
0.5	**Run3**	0.4937	0.7531	0.7377	0.7887	0.7834	0.7723	**0.7359**
	SC	0.4789	0.7407	0.7374	0.7763	0.7671	0.7641	0.7257
0.7	**Run3**	0.4816	0.7541	0.7356	0.7862	0.7806	0.7681	**0.7322**
	SC	0.4558	0.7396	0.7321	0.7694	0.7586	0.7496	0.7158

Table 2. Results of Run3 vs. SC on SemEval-2014 Test Datasets (SemEval-2015 Training dataset).

5 Conclusions and Future work

Our experiments proved that sense-based and surface-based similarity methods are complementary to each other in STS. We also realized that string kernel is more beneficial than tree kernel. Our potential future work includes: 1) enhancing our sense-based kernel approach, and 2) further enhancement in the integration between SC and our sense-based approach.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirrea, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, CO, USA.

Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. 2012. Soft Cardinality: A Parameterized Similarity Function for Text Comparison. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 449–453, Montreal, Canada.

Sergio Jimenez, George Dueñas, Julia Baquero, and Alexander Gelbukh. 2014. UNAL-NLP: Combining Soft Cardinality Features for Semantic Textual Similarity, Relatedness and Entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 732–742, Duplin, Ireland.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11): 39–41.

Roberto Navigli and Simone Paolo Ponzetto. 2010. BabelNet: Building a Very Large Multilingual Semantic Network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 216–225, Uppsala, Sweden.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. 2013. Align, Disambiguate and Walk: A Unified Approach for Measuring. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 1341–1351, Sofia, Bulgaria.

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis.* Cambridge University Press.

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.