

SXUCFN-Core: STS Models Integrating FrameNet Parsing Information

Sai Wang, Ru Li, Ruibo Wang, Zhiqiang Wang, Xia Zhang

Shanxi University, Taiyuan, China

enrique.s.wang@gmail.com

{liru, wangruibo}@sxu.edu.cn

{zhiq.wang, caesarzhangx}@163.com

Abstract

This paper describes our system submitted to *SEM 2013 Semantic Textual Similarity (STS) core task which aims to measure semantic similarity of two given text snippets. In this shared task, we propose an interpolation STS model named Model_LIM integrating FrameNet parsing information, which has a good performance with low time complexity compared with former submissions.

1 Introduction

The goal of Semantic Textual Similarity (STS) is to measure semantic similarity of two given text snippets. STS has been recently proposed by Agirre et al. (2012) as a pilot task, which has close relationship with both tasks of Textual Entailment and Paraphrase, but not equivalent with them and it is more directly applicable to a number of NLP tasks such as Question Answering (Lin and Pantel, 2001), Text Summarization (Hatzivassiloglou et al., 1999), etc. And yet, the acquiring of sentence similarity has been the most important and basic task in STS. Therefore, the STS core task of *SEM 2013 conference, is formally defined as the degree of semantic equivalence between two sentences as follows:

- **5: completely equivalent**, as they *mean the same thing*.
- **4: mostly equivalent**, but some *unimportant details differ*.

- **3: roughly equivalent**, but some *important information differs/missing*.
- **2: not equivalent**, but share some *details*.
- **1: not equivalent**, but are *on the same topic*.
- **0: on different topics**.

In this paper, we attempt to integrate semantic information into STS task besides the lower-level word and syntactic information. Evaluation results show that our STS model could benefit from semantic parsing information of two text snippets. The rest of the paper is organized as follows: Section 2 reviews prior researches on STS. Section 3 illustrates three models measuring text similarity. Section 4 describes the linear interpolation model in detail. Section 5 provides the experimental results on the development set as well as the official results on all published datasets. Finally, Section 6 summarizes our paper with direction for future works.

2 Related Work

Several techniques have been developed for STS. The typical approach to finding the similarity between two text segments is to use simple word matching method. In order to improve this simple method, Mihalcea et al. (2006) combine two corpus-based and six knowledge-based measures of word similarity, but the cost of their algorithm is expensive. In contrast, our method treats words and texts in essentially the same way.

In 2012 STS task, 35 teams participate and submit 88 runs. The two top scoring systems are UKP

and Takelab. The former system (Bär et al., 2012) uses a simple log-linear regression model to combine multiple text similarity measures (related to content, structure and style) of varying complexity. While the latter system Takelab (Šarić et al., 2012) uses a support vector regression model with multiple features measuring word-overlap similarity and syntax similarity.

The results of them score over 80%, far exceeding that of a simple lexical baseline. But both share one characteristic: they integrate lexical and syntax information without semantic information, especially FrameNet parsing information. In addition, the complexity of these algorithms is very high. Therefore, we propose a different and simple model integrating FrameNet parsing information in this paper.

3 Linear Interpolation Model

In this paper, we propose a combination interpolation model which is constructed by the results of three similarity models based on words, WordNet, FrameNet, which are called $sim_{WD}(\cdot)$, $sim_{WN}(\cdot)$ and $sim_{FN}(\cdot)$ respectively. The overall similarity $sim_{LIM}(S_1, S_2)$ between a pair of texts S_1, S_2 is computed in the following equation:

$$sim_{LIM}(S_1, S_2) = \omega_1 \cdot sim_{WD}(S_1, S_2) + \omega_2 \cdot sim_{WN}(S_1, S_2) + \omega_3 \cdot sim_{FN}(S_1, S_2) \quad (1)$$

In which, ω_1, ω_2 and ω_3 are respectively the weights of the similarity models, i.e., $\omega_1 + \omega_2 + \omega_3 = 1$; and they are all positive hyperparameters. Now, we describe the three models used in this equation.

3.1 Similarity Based on Words

This model is motivated by Vector Space Model (Salton et al., 1975). We present each sentence as a vector in the multidimensional token space. Let S_c denote the set of all words in the c -th text snippets ($c = 1, 2$); the words of bag is $W = S_1 \cup S_2$. Hence, the similarity of a pair of sentences, formally expressed as:

$$sim_{WD}(S_1, S_2) = \frac{\sum_{i=1}^{|W|} w_{1,i} \cdot w_{2,i}}{\sqrt{\sum_{i=1}^{|W|} w_{1,i}^2} \cdot \sqrt{\sum_{i=1}^{|W|} w_{2,i}^2}} \quad (2)$$

In which, we can find $w_{c,k} \in W$ ($k = 1, 2, \dots, |W|$;

$c = 1, 2$) by solving:

$$w_{c,k} = \begin{cases} 1, & \text{if } w_{c,k} \in S_c \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

From these two equations above, we can see the more identical words in a text pair, the more similar the two snippets are. Whereas, by intuition, many high-frequency functional words would not be helpful to the estimation of the similarity given in Eq.(2). Therefore, in the preprocessing stage, we compute the word frequencies per dataset, and then remove the high frequency words (top 1% in frequency list) in each segment.

3.2 Similarity Based on WordNet

This model measures semantic similarity with the help of such resources that specifically encode relations between words or concepts like WordNet (Fellbaum, 1998). We use the algorithms by Lin (1998) on WordNet to compute the similarity between two words a and b , which we call $sim_{Lin}(a, b)$. Let S_1, S_2 be the two word sets of two given text snippets, we use the method below:

$$sim_{WN}(S_1, S_2) = \frac{\sum_{i=1}^{\min(|S_1|, |S_2|)} \max(sim_{Lin}(w_{1,i}, w_{2,i}))}{\min(|S_1|, |S_2|)} \quad (4)$$

In which, $w_{c,i} \in S_c$ ($c = 1, 2$). In the numerator of Eq.(4), we try to $max(\cdot)$, $avg(\cdot)$ and $mid(\cdot)$ respectively, then we find the $max(\cdot)$ is the best.

3.3 Similarity Based on FrameNet

FrameNet lexicon (Fillmore et al., 2003) is a rich linguistic resource containing expert knowledge about lexical and predicate-argument semantics in English. In a sentence, word or phrase tokens that evoke a frame are known as **targets**. Each frame definition also includes a set of **frame elements**, or **roles**, corresponding to different aspects of the concept represented by the frame, such as participants, props, and attributes. We use the term **argument** to refer to a sequence of word tokens annotated as filling a frame role.

All the data are automatically parsed by SEMFOR¹ (Das and Smith, 2012; Das and Smith,

¹ See <http://www.ark.cs.cmu.edu/SEMAFOR/>.

2011). Figure 1 shows the parser output of a sentence pair given in Microsoft Research Video Description Corpus with annotated targets, frames and role argument pairs. It can be noticed that FrameNet parsing information could give some clues of the similarity of two given snippets and we think that integrating this information could improve the accuracy of STS task. For example, the sentences in the Figure 1 both illustrate “somebody is moving”. However, our model depends on the precision of that parser. If it would be improved, the results in STS task would be better.

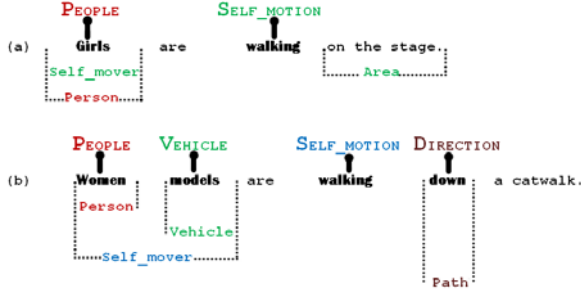


Figure 1: This is a pair of sentences in 2013 STS training data: (a) *Girls are walking on the stage*; (b) *Women models are walking down a catwalk*. The words in bold correspond to targets, which evoke semantic frames that are denoted in capital letters. Every frame is shown in a distinct color; the arguments of each frame are annotated with the same color, and marked below the sentence, at different levels; the spans marked in the block of dotted liens fulfill a specific role.

For a given sentence S_c ($c = 1, 2$) with a set of evoked frame $F_c = \langle f_1, f_2, \dots, f_n \rangle$ (n is the number of evoked frames), a set of target word with each frame $T_c = \langle t_1, t_2, \dots, t_n \rangle$ and the set of roles (namely, frame elements) $\mathcal{R}_c = \{R_{c,1}, R_{c,2}, \dots, R_{c,n}\}$, each frame contains one or more arguments $R_{c,i} = \{r_j\}$ ($i = 1, 2, \dots, n; j$ is an integer that is greater or equal to zero). Take Figure 1 as an example,

$$\begin{aligned} T_1 &= \langle \text{girls, walking} \rangle, \\ F_1 &= \langle \text{PEOPLE, SELF_MOTION} \rangle, \\ \mathcal{R}_1 &= \{R_{1,1}, R_{1,2}\}, \\ R_{1,1} &= \{\text{girls}\}, \\ R_{1,2} &= \{\text{girls, on the stage}\}; \end{aligned}$$

$$\begin{aligned} T_2 &= \langle \text{women, models, walking, down} \rangle, \\ F_2 &= \langle \text{PEOPLE, VEHICLE,} \\ &\quad \text{SELF_MOTION, DIRECTION} \rangle, \end{aligned}$$

$$\begin{aligned} \mathcal{R}_2 &= \{R_{2,1}, R_{2,2}, R_{2,3}, R_{2,4}\}, \\ R_{2,1} &= \{\text{women}\}, R_{2,2} = \{\text{models}\}, \\ R_{2,3} &= \{\text{women models}\}, R_{2,4} = \{\text{down}\}. \end{aligned}$$

In order to compute $sim_{Fr}(\cdot)$ simply, we also use an interpolation model to combine the similarities based on target words $sim_{Tg}(\cdot)$, frames $sim_{Fr}(\cdot)$ and frame relations $sim_{Re}(\cdot)$. They are estimated as the following:

When computing the similarity on target word level $sim_{Tg}(S_1, S_2)$, we also consider each sentence as a vector of target words as is seen in Eq.(5).

$$\begin{aligned} T &= T_1 \cup T_2; \\ sim_{Tg}(S_1, S_2) &= \frac{\sum_{i=1}^{|T|} t_{1,i} \cdot t_{2,i}}{\sqrt{\sum_{i=1}^{|T|} t_{1,i}^2} \cdot \sqrt{\sum_{i=1}^{|T|} t_{2,i}^2}} \end{aligned} \quad (5)$$

In which, we can find $t_{c,k} \in T$ ($k = 1, 2, \dots, |T|$; $c = 1, 2$) by solving:

$$t_{c,k} = \begin{cases} 1, & \text{if } f_{c,j} \in F_c \text{ and } t_{c,k} \in T_c \\ & (j = 1, 2, \dots, |F|) \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

Let $sim_{Fr}(S_1, S_2)$ be the similarity on frame level as shown in Eq.(7), with each sentence as a vector of frames. We define $f_{1,i}, f_{2,i}$ like $w_{c,k}$ in Eq.(3).

$$\begin{aligned} F &= F_1 \cup F_2; \\ sim_{Fr}(S_1, S_2) &= \frac{\sum_{i=1}^{|F|} f_{1,i} \cdot f_{2,i}}{\sqrt{\sum_{i=1}^{|F|} f_{1,i}^2} \cdot \sqrt{\sum_{i=1}^{|F|} f_{2,i}^2}} \end{aligned} \quad (7)$$

Before computing the role relationship between the pair of sentences, we should find the **containment relationship** of each pair of frames in one sentence. We use a rule to define the containment relationship:

Given two frames $f_{c,i}, f_{c,j}$ in a sentence S_c , if $t_{c,j} \in R_{c,j}$ ($i \neq j$), then $f_{c,j}$ contains $f_{c,i}$ - and that is $f_{c,i}$ is a child of $f_{c,j}$. After that we add them into the set of frame relationship $Rlt_c = \langle \langle f_{c,i}^k, f_{c,j}^k \rangle \rangle_{k=0}^N = \langle rel_{c,k} \rangle_{k=0}^N$, ($k \geq 0$).

We consider the relationship between two frames in a sentence as a 2-tuple, and again use Figure 1 as an example,

$$\begin{aligned} Rlt_1 &= \langle \langle \text{PEOPLE, SELF_MOTION} \rangle \rangle; \\ Rlt_2 &= \langle \langle \text{PEOPLE, SELF_MOTION}, \\ &\quad \langle \text{VEHICLE, SELF_MOTION} \rangle \rangle. \end{aligned}$$

Besides, we do exactly the same with both frames, namely $rel_{c,i} \in Rlt_c$ ($c = 1,2$) the value of $rel_{c,i}$ is 1. The similarity on frame relationship level $sim_{Re}(S_1, S_2)$ presents each sentence as a vector of roles as shown in Eq.(8).

$$Rlt = Rlt_1 \cup Rlt_2; \quad (8)$$

$$sim_{Re}(S_1, S_2) = \frac{\sum_{i=1}^{|Rlt|} rel_{1,i} \cdot rel_{2,i}}{\sqrt{\sum_{i=1}^{|Rlt|} rel_{1,i}^2} \cdot \sqrt{\sum_{i=1}^{|Rlt|} rel_{2,i}^2}}$$

Lastly, the shallow semantic similarity between two given sentences is computed as:

$$Sim_{FN}(S_1, S_2) = \alpha \cdot sim_{Tg}(S_1, S_2) + \beta \cdot sim_{Fr}(S_1, S_2) + \gamma \cdot sim_{Re}(S_1, S_2) \quad (9)$$

In which, $\alpha + \beta + \gamma = 1$, and they are all positive hyperparameters. As shown in Figure 2, we plot the Pearson correlation (vertical axis) against the combination of parameters (horizontal axis) in all 2013 STS train data (2012 STS data). We notice that generally the Pearson correlation is fluctuates, and the correlation peak is found at 32, which in Table 1 is $\alpha=0.6, \beta=0.3, \gamma=0.1$.

ID	α	β	γ	ID	α	β	γ	ID	α	β	γ
1	1	0	0	23	0.7	0.2	0.1	45	0	0.4	0.6
2	0.9	0	0.1	24	0.6	0.2	0.2	46	0.5	0.5	0
3	0.8	0	0.2	25	0.5	0.2	0.3	47	0.4	0.5	0.1
4	0.7	0	0.3	26	0.4	0.2	0.4	48	0.3	0.5	0.2
5	0.6	0	0.4	27	0.3	0.2	0.5	49	0.2	0.5	0.3
6	0.5	0	0.5	28	0.2	0.2	0.6	50	0.1	0.5	0.4
7	0.4	0	0.6	29	0.1	0.2	0.7	51	0	0.5	0.5
8	0.3	0	0.7	30	0	0.2	0.8	52	0.4	0.6	0
9	0.2	0	0.8	31	0.7	0.3	0	53	0.3	0.6	0.1
10	0.1	0	0.9	32	0.6	0.3	0.1	54	0.2	0.6	0.2
11	0	0	1	33	0.5	0.3	0.2	55	0.1	0.6	0.3
12	0.9	0.1	0	34	0.4	0.3	0.3	56	0	0.6	0.4
13	0.8	0.1	0.1	35	0.3	0.3	0.4	57	0.3	0.7	0
14	0.7	0.1	0.2	36	0.2	0.3	0.5	58	0.2	0.7	0.1
15	0.6	0.1	0.3	37	0.1	0.3	0.6	59	0.1	0.7	0.2
16	0.5	0.1	0.4	38	0	0.3	0.7	60	0	0.7	0.3
17	0.4	0.1	0.5	39	0.6	0.4	0	61	0.2	0.8	0
18	0.3	0.1	0.6	40	0.5	0.4	0.1	62	0.1	0.8	0.1
19	0.2	0.1	0.7	41	0.4	0.4	0.2	63	0	0.8	0.2
20	0.1	0.1	0.8	42	0.3	0.4	0.3	64	0.1	0.9	0
21	0	0.1	0.9	43	0.2	0.4	0.4	65	0	0.9	0.1
22	0.8	0.2	0	44	0.1	0.4	0.5	66	0	1	0

Table 1: Different combinations of α, β, γ ($\alpha + \beta + \gamma = 1$) with ID that is horizontal axis in Figure 2. This table also applies to different combinations of $\omega_1, \omega_2, \omega_3$ ($\omega_1 + \omega_2 + \omega_3 = 1$) with ID that is horizontal axis in Figure 3.

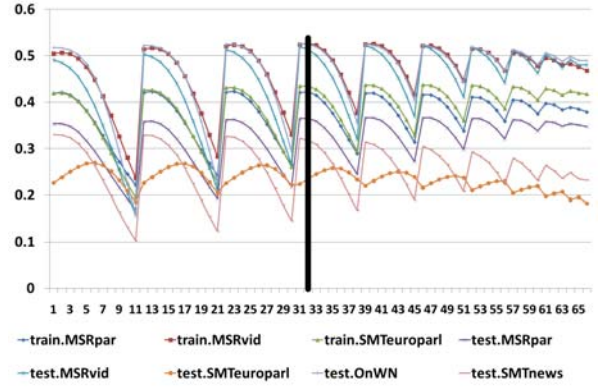


Figure 2: This graph shows the variation of Pearson correlation (vertical axis) in all 2013 STS train data (2012 STS data), with numbers (horizontal axis) indicating different combinations α, β, γ in Table 1 and when the value of result confidence is 100. The effect values are represented by a vertical line (i.e. ID = 32).

4 Tuning Hyperparameters

Eq.(1) is a very simple linear interpolation model, and we tune the hyperparameters on the whole 2012 STS data.

As shown in Figure 3, we plot the Pearson correlation (vertical axis) for the different combination of parameters ω_1, ω_2 and ω_3 (horizontal axis). We notice that generally the Pearson correlation fluctuates with a dropping tendency in most cases, and the correlation peak presents at 13, which in Table 1 is $\omega_1=0.8, \omega_2=0.1, \omega_3=0.1$.

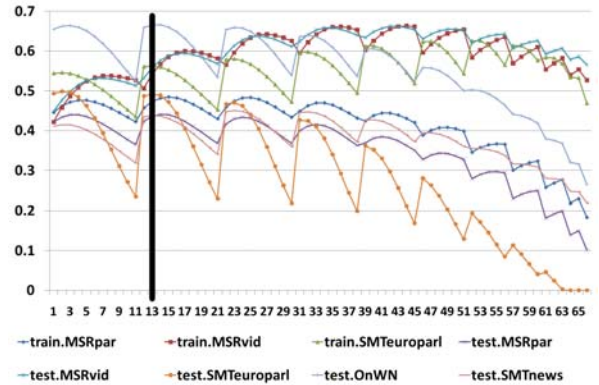


Figure 3: This graph shows the variation of Pearson correlation (vertical axis) in all 2013 STS train data (2012 STS data), with numbers (horizontal axis) indicating different combinations $\omega_1, \omega_2, \omega_3$ in Table 1 and when the value of result confidence is 100. The effect values are represented by a vertical line (i.e. ID = 13).

5 Results

We submit four runs: the first one (Model_WD) is based on word similarity; the second one (Model_WN) which is only using the similarity based on WordNet, is submitted with the team name of SXULLL; the third one (Model_FN) which uses FrameNet similarity defined in Section 3.3; and the last one in which we combine the three similarities described in Section 4 together with an interpolation model. In addition, we map our outputs multiply by five to the [0-5] range.

It is worth notice that in the first model, we lowercase all words and remove all numbers and punctuations. And in the third model, we extract all frame-semantic roles with SEMFOR.

In the experiment, we use eight datasets totally - namely MSRpar, MSRvid, SMTEuroparl, OnWN, SMTnews, headlines, FNWN and SMT - with their gold standard file to evaluate the performance of the submitted systems. Evaluation is carried out using the official scorer which computes Pearson correlation between the human rated similarity scores and the system’s output. The final measure is the score that is weighted by the number of text pairs in each dataset (“Mean”). See Agirre et al. (2012) for a full description of the metrics.

5.1 Experiments on STS 2012 Data

There is no new train data in 2013, so we use 2012 data as train data. From Table 2, 3 we can see that the Model_LIM has better performance than the other three models.

	MSRpar	MSRvid	SMTEuroparl	Mean
Model_WD	0.4532	0.4487	0.6467	0.5153
Model_WN	0.2718	0.5410	0.6225	0.4774
Model_FN	0.4437	0.5530	0.5178	0.5048
Model_LIM	0.4896	0.5533	0.6681	0.5696

Table 2: Performances of the four models on 2012 train data. The highest correlation in each column is given in bold.

From Table 2, we notice that all the models except Model_FN, are apt to handle the SMTEuroparl that involves long sentences. For Model_FN, it performs well in computing on short and similarly structured texts such as MSRvid (This will be confirmed in test data later). Although WordNet and FrameNet model has a mere weight of 20% in Model_LIM (i.e. $\omega_1 + \omega_2 = 0.2$), the run which integrate more semantic information displays a con-

sistent performance across the three train sets (especially in SMTEuroparl, the Pearson correlation rises from 0.5178 to 0.66808), when compared to the other three.

	MSRpar	MSRvid	SMTEuroparl	OnWN	SMTnews	Mean
Baseline	0.4334	0.2996	0.4542	0.5864	0.3908	0.4356
Model_WD	0.4404	0.5464	0.5059	0.6751	0.4583	0.5346
Model_WN	0.1247	0.6608	0.0637	0.4089	0.3436	0.3417
Model_FN	0.3830	0.6082	0.3537	0.6091	0.4061	0.4905
Model_LIM	0.4489	0.6301	0.5086	0.6841	0.4872	0.5631
UKP_run2	0.6830	0.8739	0.5280	0.6641	0.4937	0.6773

Table 3: Performances of our three models as well as the baseline and UKP_run2 (that is ranked 1 in last STS task) results on 2012 test data. The highest correlation in each column is given in bold.

The 2012 STS test results obtained by first ranking UKP_run2 and baseline system are shown in Table 3, it is interesting to notice that performance of Model_WD is similar with Model_LIM except on MSRvid, the text segments in which there are fewer identical words because of the semantic equivalence. For Model_FN, we can see it performs well on short and similarly structured texts (MSRvid and OnWN) as mentioned before. This is because the precision of FrameNet parser took effect on the FrameNet-based models performance. Compared to UKP_run2, the performance of Model_LIM is obviously better on OnWN set, while on SMTEuroparl and SMTnews this model scores slightly lower than UKP_run2. Finally, Model_LIM did not perform best on MSRpar and MSRvid compared with UKP_run2, but it has low time complexity and integrates semantic information.

5.2 Official Results on STS 2013 Test Data

Table 4 provides the official results of our submitted systems, along with the rank on each dataset. Generally, all results outperform the baseline, based on simple word overlap. However, the performance of Model_LIM is not always the best in the three runs for each dataset. From the table we can note that a particular model always performs well on the dataset including the lexicon on which the model is based on e.g. Model_WN in OnWN, Model_FN in FNWN. Besides, Model_WD and Model_LIM almost have same scores except in OnWN set, because in Model_LIM is included with WordNet resource.

	headlines	OnWN	FNWN	SMT	Mean
Baseline	0.5399 (66)	0.2828 (80)	0.2146 (66)	0.2861 (65)	0.3639 (73)
Model_WD	0.6806 (24)	0.5355 (44)	0.3181 (48)	0.3980 (4)	0.5198 (27)
Model_WN	0.4840 (78)	0.7146 (12)	0.0415 (83)	0.1543 (86)	0.3944 (69)
Model_FN	0.4881 (76)	0.6146 (27)	0.4237 (9)	0.3844 (6)	0.4797 (46)
Model_LIM	0.6761 (29)	0.6481 (23)	0.3025 (51)	0.4003 (3)	0.5458 (14)

Table 4: Performances of our systems as well as baseline on STS 2013 individual test data, accompanied by their rank (out of 90) shown in brackets. Scores in bold denote significant improvements over the baseline.

As seen from the system rank in table, the optimal runs in the three submitted system remain with Model_LIM. Not only Model_LIM performs best on two occasions, but also Model_FN ranks top ten twice, in FNWN and SMT respectively, we owe this result to the contribution of FrameNet parsing information.

6 Conclusion

We have tested all the models on published STS datasets. Compared with the official results, Model_LIM system is apt to handle the SMT that involves long sentences. Moreover, this system just integrates words, WordNet and FrameNet semantic information, thus it has low time complexity. There is still much room for improvement in our work. For example, we will attempt to use multivariate regression software to tuning the hyperparameters.

Acknowledgments

This work is supported by the National Nature Science Foundation of China (No.60970053), by the National High-tech Research and Development Projects (863) grant No.2006AA01Z142, by the State Language Commission of China No.YB125-19 as well as by the International Cooperation of Shanxi Province, Contracts 2010081044. And we would like to thank the organizer for the tremendous effort they put into formulating this challenging work.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation, in conjunction with the 1st Joint Conference on Lexical and Computational Semantics*, 385–393.

Dekang Lin, Patrick Pantel. 2001. Discovery of Inference Rules for Question Answering. *Natural Language Engineering*, 7(4):343-360.

Vasileios Hatzivassiloglou, Judith L. Klavans, and Eleazar Eskin. 1999. Detecting Text Similarity over Short Passages: Exploring Linguistic Feature Combinations via Machine Learning. In *proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 224-231.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and Knowledge-based Measures of Text Semantic Similarity. In *Proceedings of the National Conference on Artificial Intelligence*, 21(1): 775-780.

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, in conjunction with the 1st Joint Conference on Lexical and Computational Semantics*, 435-440.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. TakeLab: Systems for Measuring Semantic Text Similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation, in conjunction with the 1st Joint Conference on Lexical and Computational Semantics*, 441-448.

G. Salton, A. Wong, C.S. Yang. 1975. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11):613-620.

C. J. Fillmore, C. R. Johnson and M. R.L. Petruck. 2003. Background to FrameNet. *International Journal of Lexicography*, 16: 235-250.

Dipanjan Das and Noah A. Smith. 2012. Graph-Based Lexicon Expansion with Sparsity-Inducing Penalties. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 677-687.

Dipanjan Das and Noah A. Smith. 2011. Semi-Supervised Frame-Semantic Parsing for Unknown Predicates. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, 1435-1444.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning*, 296-340.