BIT at SemEval-2017 Task 1: Using Semantic Information Space to Evaluate Semantic Textual Similarity

Hao Wu, Heyan Huang*, Ping Jian, Yuhang Guo, Chao Su
Beijing Engineering Research Center of High Volume Language Information Processing
and Cloud Computing Applications, School of Computer Science,
Beijing Institute of Technology, Beijing, China
{wuhao123, hhy63, pjian, guoyuhang, suchao}@bit.edu.cn

Abstract

This paper presents three systems for semantic textual similarity (STS) evaluation at SemEval-2017 STS task. One is an unsupervised system and the other two are supervised systems which simply employ the unsupervised one. All our systems mainly depend on the *semantic information space* (SIS), which is constructed based on the semantic hierarchical taxonomy in WordNet, to compute non-overlapping information content (IC) of sentences. Our team ranked 2nd among 31 participating teams by the primary score of Pearson correlation coefficient (PCC) mean of 7 tracks and achieved the best performance on Track 1 (AR-AR) dataset.

1 Introduction

Given two snippets of text, semantic textual similarity (STS) measures the degree of equivalence in the underlying semantics. STS is a basic but important issue with multitude of application areas in natural language processing (NLP) such as example based machine translation (EBMT), machine translation evaluation, information retrieval (IR), question answering (QA), text summarization and so on.

The SemEval STS task has become the most famous activity for STS evaluation in recent years and the STS shared task has been held annually since 2012 (Agirre et al., 2012, 2013, 2014, 2015, 2016; Cer et al., 2017), as part of the SemEval/*SEM family of workshops. The organizers have set up publicly available datasets of sentence pairs with similarity scores from human annotators, which are up to more than 16,000 sentence pairs for training and evaluation, and attracted a large number of teams with a variety of systems to participate the competitions.

Generally, STS systems could be divided into two categories: One kind is unsupervised systems (Li et al., 2006; Mihalcea et al., 2006; Islam and Inkpen, 2008; Han et al., 2013; Sultan et al., 2014b; Wu and Huang, 2016), some of which are appeared for a long time when there wasn't enough training data; The other kind is supervised systems (Bär et al., 2012; Šarić et al., 2012; Sultan et al., 2015; Rychalska et al., 2016; Brychcín and Svoboda, 2016) applying machine learning algorithms, including deep learning, after adequate training data has been constructed. Each kind of methods has its advantages and application areas. In this paper, we present three systems, one unsupervised system and two supervised systems which simply make use of the unsupervised one.

2 Preliminaries

Following the standard argumentation of information theory, Resnik (1995) proposed the definition of the information content (IC) of a concept as follows:

$$IC\,(c) = -\log P(c), \tag{1}$$

where $P(c)$ refers to statistical frequency of concept c.

Since information content (IC) for multiple words, which sums the non-overlapping concepts IC, is a computational difficulties for knowledge based methods. For a long time, IC related methods were usually used as word similarity (Resnik, 1995; Jiang and Conrath, 1997; Lin, 1997) or word weight (Li et al., 2006; Han et al., 2013) rather than the core evaluation modules of sentence similarity methods (Wu and Huang, 2016).

*Corresponding author

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 77–84,
Vancouver, Canada, August 3 - 4, 2017. ©2017 Association for Computational Linguistics

2.1 STS evaluation using SIS

To apply non-overlapping IC of sentences in STS evaluation, we construct the semantic information space (SIS), which employs the super-subordinate (is-a) relation from the hierarchical taxonomy of WordNet (Wu and Huang, 2016). The space size of a concept is the information content of the concept. SIS is not a traditional orthogonality multidimensional space, while it is the space with inclusion relation among concepts. Sentences in SIS are represented as a real physical space instead of a point in vector space.

We have the intuitions about similarity: The similarity between A and B is related to their commonality and differences, the more commonality and the less differences they have, the more similar they are; The maximum similarity is reached when A and B are identical, no matter how much commonality they share(Lin, 1998). The principle of Jaccard coefficient (Jaccard, 1908) is accordance with the intuitions about similarity and we define the similarity of two sentences S_a and S_b based on it:

$$sim\left(s_a, s_b\right) = \frac{IC\left(s_a \cap s_b\right)}{IC\left(s_a \cup s_b\right)}. \tag{2}$$

The quantity of the intersection of the information provided by the two sentences can be obtained through that of the union of them:

$$IC\left(s_a \cap s_b\right) = IC\left(s_a\right) + IC\left(s_b\right) - IC\left(s_a \cup s_b\right). \tag{3}$$

So the remaining problem is how to compute the quantity of the union of non-overlapping information of sentences. We calculate it by employing the inclusion-exclusion principle from combinatorics for the total IC of sentence s_a and the same way is used for sentence s_b and both sentences:

$$\begin{aligned} IC\left(s_a\right) &= IC\left(\bigcup_{i=1}^{n} c_i^a\right) \\ &= \sum_{k=1}^{n} (-1)^{k-1} \sum_{1 \le i_1 < \cdots < i_k \le n} IC\left(c_{i_1}^a \cap \cdots \cap c_{i_k}^a\right). \end{aligned} \tag{4}$$

For the IC of n-concepts intersection in Equation (4), we use the following equation[1]:

```
Algorithm 1: getInExTotalIC(S)
  Input: $S$ : $\{c_i | i = 1, 2, \ldots, n; n = |S|\}$
  Output: $tIC$: Total IC of input $S$
  if $S = \varnothing$ then
      return 0
  Initialize: $tIC \leftarrow 0$
  for $i = 1; i \le n; i{+}{+}$ do
      foreach comb in $C(n, i)$-combinations do
          $cIC \leftarrow commonIC(comb)$
          $tIC {+}= (-1)^{i-1} \cdot cIC$
  return $tIC$
```

$$\begin{aligned} commonIC\left(c_1, \cdots, c_n\right) &= IC\left(\bigcap_{i=1}^{n} c_i\right) \\ &\approx \max_{c \in subsum(c_1, \cdots, c_n)} \left[-\log P(c)\right], \end{aligned} \tag{6}$$

where, $subsum\left(c_1, \cdots, c_n\right)$ is the set of concepts that subsume all the concepts of $c_1, \cdots, c_n$ in SIS.

Algorithm 1 is according to Equation (4) and (6), here $C\left(n, i\right)$ is the number of combinations of i-concepts from n-concepts, $commonIC(comb)$ is calculated through Equation (6).

For more details about this section, please see the paper (Wu and Huang, 2016) for reference.

2.2 The Efficient Algorithm for Sentence IC

According to the Binomial Theorem, the amount of combinations for $commonIC(comb)$ calculation from Equation (4) is:

$$C\left(n, 1\right) + \cdots + C\left(n, n\right) = 2^n - 1. \tag{7}$$

Searching subsumers in the hierarchical taxonomy of WordNet is the most time-consuming operation. Define one time searching between concepts be the minimum computational unit. Considering searching subsumers among multiple concepts, the real computational complexity is more than $0 * C(n, 1) + 1 * C\left(n, 2\right) + \cdots + (n-1) * C\left(n, n\right)$.

Note that the computational complexity through the inclusion-exclusion principle is more than $O(2^n)$. To decrease the computational complexity, we exploit the efficient algorithm for precise non-overlapping IC computing of sentences by making use of the thinking of the different set in hierarchical network (Wu and Huang,

[1] For the sake of high computational complexity introduced by Equation (4), we simplify the calculation of common IC of n-concepts and use the approximate formula in Equation (6). The accurate formula of common IC is:

$$commonIC\left(c_1, \cdots, c_n\right) = IC\left(\bigcap_{i=1}^{n} c_i\right) = IC\left(\bigcup_{j=1}^{m} c_j\right), \tag{5}$$

where $c_j \in subsum\left(c_1, \cdots, c_n\right)$, m is the total number of c_j. We could see Equation.(4) and (5) are indirect recursion.

Algorithm 2: $getTotalIC(S)$

```
Input: $S : \{c_i|i = 1, 2, \ldots, n; n = |S|\}$
Output: $tIC$: Total IC of input $S$
if $S = \varnothing$ then
    return 0
Initialize: $tIC \leftarrow 0, Root(0) \leftarrow \varnothing$
for $i = 1; i \leq n; i++$ do
    $Intersect(i|i-1), Root(i) \leftarrow getIntersect(c_i, Root(i-1))$
    $ICG \leftarrow IC(c_i) - getTotalIC(Intersect(i|i-1))$
    $tIC+ = ICG$
return $tIC$
```

Algorithm 3: $getIntersect(c_i, Root(i-1))$

```
Input: $c_i$, $Root(i-1)$
Output: $Intersect(i|i-1)$, $Root(i)$
Initialize: get $Root(c_i)$ from WordNet
$Intersect(i|i-1) \leftarrow \varnothing; Root(i) \leftarrow Root(i-1)$
if $Root(i) = \varnothing$ then                    /* $i = 1$ */
    $Root(i) \leftarrow Root(c_i)$
    return $Intersect(i|i-1), Root(i)$
foreach $r_i \in Root(c_i)$ do
    $pos \leftarrow depth(r_i) - 1$ /* $pos$ ⇔ root */
    foreach $r_{i-1} \in Root(i-1)$ do
        $(p, q) \leftarrow$ deepest common node position: $p$ in $r_i$, $q$ in $r_{i-1}$
        if $p = 0$ then              /* $r_i$ in $r_{i-1}$ */
            add $c_i$ to $Intersect(i|i-1)$
            break the outer foreach loop
        if $q = 0$ then              /* $r_{i-1}$ in $r_i$ */
            remove $r_{i-1}$ from $Root(i)$
        if $p < pos$ then /* $r_{i-1}$ intersect at deeper node in $r_i$ */
            $pos \leftarrow p$
    add $r_i$ to $Root(i)$
    add $c_{pos} \in r_i$ to $Intersect(i|i-1)$
return $Intersect(i|i-1), Root(i)$
```

2017): We add the words into the SIS one by one each time and sum the gain IC of $ICG(c_i)$ from the newly added concept c_i. For sentence $S = \{c_i|i = 1, 2, \ldots, n; n = |S|\}$, where c_i is the concept of the i-th concept in S, $|S|$ is concept count of S, the formula of $ICG(c_i)$ is as follows:

$$IC(S) = \sum_{i=1}^{n} ICG(c_i) \tag{8}$$

For convenience in the expression of $ICG(c_i)$, we define some functions: $Root(c_i)$ indicates the set of paths, each path is the node list **from $\mathbf{c_i}$ to the root** in the nominal hierarchical taxonomy of WordNet. $Root(n)$ is the short form of $Root(c_1, \cdots, c_n)$. Formally, let $Set(p)$ be the set of nodes in path p, $Root(n) = \{p_k|\forall p_k \in Root(c_i), \nexists p_t \in Root(c_j), Set(p_k) \subseteq Set(p_t).i = 1, 2, \ldots, n; j = 1, 2, \ldots, n\}$. $|Root(c_i)|$ means the number of paths in $Root(c_i)$. $HSN(c_i)$ expresses the set of nodes in any of path in $Root(c_i)$. $HSN(n)$ is the short form of $HSN(c_1, \cdots, c_n)$, formally, $HSN(n) = \{c_k|c_k \in HSN(c_i).i = 1, 2, \ldots, n\}$.

Let $depth(c)$ be the max depth from concept c to the root. We define $Intersect(n+1|n) = \{c_i|\forall c_i \in \{Set(p_t) \wedge HSN(n)\}, \nexists c_j \in \{Set(p_t) \wedge HSN(n)\}, depth(c_i) \leq depth(c_j).p_t \in Root(c_{n+1}); t = 1, \cdots, |Root(c_{n+1})|\}$ and $totalIC(c_1, \cdots, c_n)$ is the quantity of total information of n-concepts. We have

$$ICG(C_i) = IC(c_i) - totalIC(Intersect(i|i-1)). \tag{9}$$

Algorithm 2 and 3 are according to Equation (8) and (9). Algorithm 3 is approximately equal to one time subsumer searching between concepts, thus the computational complexity of Algorithm 2 is $O(n)$. This indicates SIS methods could be applied to any length of sentences even short paragraphs. The open source implementations of Algorithm 2 and 3 with related library are also available at GitHub[2].

Theoretical system with lemmas and theorems has been established for supporting the correctness of Equation (8) and (9). For more details about this section, please see the paper (Wu and Huang, 2017) for reference.

2.3 Increasing Word Recall Rate for SIS

We made three aspects improvements in our another previous work:

First, we utilize WordNet to directly obtain the nominal forms of a content word which is not a noun mainly through derivational pointers in WordNet. The word formation helps enhance the recall rate of known content words in sentence-to-SIS mappings. Second, name entity (NE) recognition tool (Manning et al., 2014) and the alignment

[2] https://github.com/hao123wu/STS

tool (Sultan et al., 2014a) are employed to obtain non-overlapping unknown NEs, which are used for simulating non-overlapping IC in SIS. The alignment tool is mainly used for finding actually same NEs with different string forms and inconsistent NE annotations by the NE recognition tool. Through the statistic values of known NEs of the same kinds from previous datasets, we simulate the IC of out-of-vocabulary NEs in SIS. Finally, sentence IC is augmented by word weights which could deem as the importance of words.

The above contents of this subsection is mainly based on the work which is currently under review.

3 System Overview

We submitted three systems: One is the unsupervised system of exploiting non-overlapping IC in SIS, the other two are supervised systems of making use of the methods of sentence alignment and word embedding respectively.

3.1 Preprocessing

First of all, we translated all the other languages into English by employing Google machine translation system[3] and preprocessed the test datasets with *tokenizer.perl* and *truecase.perl*, which are the tools from Moses machine translation toolkit (Koehn et al., 2007), then utilized the preprocessed datasets to do POS obtaining and lemmatization by utilizing NLTK (Bird, 2006), and finally made use of lemma to do sentence alignment (Sultan et al., 2014a) and name entity recognition (Manning et al., 2014). We use the lemma instead of the original word in all the situations where need words to participate for the consideration of simplicity.

We also developed a word spelling correction module based on Levenshtein distance which is special for the spelling mistakes in STS datasets. It proved important for the eventual performances in previous years, however, it was not so critical this year.

3.2 Run 1: Unsupervised SIS

Run 1 is from the unsupervised system constructed using the framework described in Section 2 and the implementation is as follows:

Word IC calculation employs Equation (1) and the probability of a concept c is:

$$P(c) = \frac{\sum_{n \in words(c)} count(n)}{N} \tag{10}$$

where $words(c)$ is the set of all the words contained in concept c **and its sub-concepts** in WordNet, N is the sum of frequencies of words contained in all the concepts in the hierarchy of semantic net. The word statistics are from British National Corpus (BNC) obtained by NLTK (Bird, 2006). Sentence IC computation applies Equation (9).

For the simplification, we choose the concept of a word with the minimal IC, which denotes the most common sense of a word, in all the circumstances of conversion of word-to-concept and the selection between two aligned words, instead of word sense disambiguation (WSD).

3.3 Run 2: Supervised IC and Alignment

As the aligner of Sultan et al. (2014a) is successfully applied in STS evaluation, we should leverage its advantage of finding potential word aligned pairs from both sentences, especially for different surface forms. However, we did not obtain the global inverse document frequency (IDF) data on time, thus we did not employ the aligner of Brychcín and Svoboda (2016), which is the improved version of Sultan et al. (2014a), that introduces IDF information of words in the similarity formula.

In this run, we use support vector machines (Chang and Lin, 2011) (SVM) for regression, more specifically sequential minimal optimization (Shevade et al., 2000) (SMO). There two features: One is the output of SIS, the other is that of unsupervised method of Sultan et al. (2015).

Actually, we tested some other regression methods. We found that LR and SVM always outperform the others. The tool for regression methods are implemented in WEKA (Hall et al., 2009).

3.4 Run 3: Supervised IC and Embeddings

Deep learning has become a hot topic in recent years and many supervised methods of STS incorporate deep learning models. At SemEval 2016 STS task, at least top 5 teams included deep learning modules according to incomplete statistics (Agirre et al., 2016).

In this run, we take advantage of the embeddings that obtained information from large scale

[3] http://translate.google.com

Track	Dataset	Total	GS Pairs
Track 1	Arabic-Arabic	250	250
Track 2	Arabic-English	250	250
Track 3	Spanish-Spanish	250	250
Track 4a	Spanish-English	250	250
Track 4b	Spanish-English-WMT	250	250
Track 5	English-English	250	250
Track 6	English-Turkish	500	250
Sum		2000	1750

Table 1: Test sets at SemEval 2017 STS task.

corpora and train the linear regression (LR) model. There two features: One is the outputs of SIS, the other is from a modified version of basic sentence embedding which is the simply combination of word embeddings.

The word embedding vectors are generated from word2vec (Mikolov et al., 2013) over the 5th edition of the Gigaword (LDC2011T07) (Parker et al., 2011). We also preprocess the Gigaword data with *tokenizer.perl* and *truecase.perl*. We modify this basic sentence embedding by importing domain IDF information. The domain IDFs of words could be obtained from the current test dataset by deeming each sentence as a document. We did not directly use the domain IDFs d as the weight of a word embedding. On previous datasets, we found $d^{0.8}$ as its weight performed nearly the best.

4 Data

SemEval 2017 STS task assesses the ability of systems to determine the degree of semantic similarity between monolingual and cross-lingual sentences in Arabic, English, Spanish and a surprise language of Turkish. The shared task is organized into a set of secondary sub-tracks and a single combined primary track. Each secondary sub-track involves providing STS scores for monolingual sentence pairs in a particular language or for cross-lingual sentence pairs from the combination of two particular languages. Participation in the primary track is achieved by submitting results for all of the secondary sub-tracks (Cer et al., 2017).

As shown in Table 1, the SemEval 2017 STS shared task contains 1750 pairs with gold standard (GS) out of total 2000 pairs from 7 different tracks. Systems were required to annotate all the pairs and performance was evaluated on all pairs or a subset with GS in the datasets. The GS for each pair ranges from 0 to 5, with the values indicating the corresponding interpretations: 5 indicates completely equivalence; 4 expresses mostly equivalent with differences only in some unimportant details; 3 means roughly equivalent but with differences in some important details; 2 means non-equivalence but sharing some details; 1 means the pairs only share the same topic; and 0 represents no overlap in similarity.

5 Evaluation

The evaluation metric is the Pearson product-moment correlation coefficient (PCC) between semantic similarity scores of machine assigned and human judgements. PCC is used for each individual test set, and the primary evaluation is measured by weighted mean of PCC on all datasets (Cer et al., 2017).

Performances of our three runs on each of SemEval 2017 STS test set are shown in Table 2. Bold numbers represents the best scores from any our system on each test set, including the primary scores. *Cosine Baseline* utilizes basic sentence embedding method for monolingual similarity (Track 1, 3 and 5) provided officially by STS organizers; *Best system* denotes all the scores are from the state-of-the-art system; *All Systems Best* means the best scores from all the systems participated in each track, regardless of whether they come from the same system; *Differences* indicates the differences between the best scores from our three systems and *All Single Best* in each track, primary difference is between our best system and state-of-the-art system. *Team Rankings* show the rankings of our best scores from that of other teams. *Team Rankings* of Primary could be the most important ranking for participants who submitted scores for all tracks.

Our team ranked 2nd for the primary score and achieved the best performance in Track 1 (Arabic-Arabic). Track 1 is the only track that totally employed new languages which has no references from the past (cross-lingual evaluation contains English sentences).

The very failing performance is in Track 4b. We guess the reasons could be the followings and further research is needed on this issue:

1) Our methods, especially for unsupervised SIS, ignore some important information as the embedding methods and are currently not suit for complicated post-editing sentences. We tested basic sentence embedding method in isolation which could achieve the score of more than 0.16,

	Primary	Track 1	Track 2	Track 3	Track 4a	Track 4b	Track 5	Track 6
Run 1	0.6703	0.7535	**0.7007**	0.8323	0.7813	0.0758	0.8161	**0.7327**
Run 2	0.6662	**0.7543**	0.6953	0.8289	0.7761	0.0584	0.8222	0.7280
Run 3	**0.6789**	0.7417	0.6965	**0.8499**	**0.7828**	**0.1107**	**0.8400**	0.7305
Cosine Baseline	0.5370	0.6045	0.5155	0.7117	0.6220	0.0320	0.7278	0.5456
Best System	0.7316	0.7440	0.7493	0.8559	0.8131	0.3363	0.8518	0.7706
All Single Best	-	0.7543	0.7493	0.8559	0.8302	0.3407	0.8547	0.7706
Differences	5.3%	-0.8%	4.9%	0.6%	4.7%	23.0%	1.5%	3.8%
Team Rankings	2	1	2	2	3	14	4	2

Table 2: Performances on SemEval 2017 STS evaluation datasets.

much better than our IC based systems of Run 1 (0.0758) and Run 2 (0.0584),which are without embedding modules.

2) The translation quantity for long sentences by machine translation may be not good enough as that for short sentences. The translation results may lose some information in the original sentences for SIS and introduce more noise.

6 STS benchmark

In order to provide a standard benchmark to compare among the state-of-the-art in Semantic Textual Similarity for English, the organizers of SemEval STS tasks are already setting a leaderboard this year which includes results of some selected systems. The benchmark comprises a selection of the English datasets used in the STS tasks in the context of SemEval from 2012 to 2017 and it is organized into train, development and test (Cer et al., 2017).

Our systems are selected by the organizers to submit the results for STS benchmark. We employ the models that described above, but a small difference is in Run 3: $d^{0.9}$ was used as the weights of word embeddings, which could achieve the best performance of cosine similarity from the summed word embeddings in isolation. As our models need not tune hyperparameters, the train part is used for tuning parameters and training models while the development part and the test part are used for the testing of the final systems. Table 3 shows the performances of our systems.

From the table we could see Run 3 provides the best performance in benchmark, which is in accordance with the results in SemEval 2017 STS task. Our best system ranks 2nd at present. More details about STS benchmark and the real-time leaderboard could be find in the official website[4].

Set	Size	Run 1	Run 2	Run 3
Development	1500	0.8194	0.8240	**0.8291**
Test	1379	0.7942	0.7962	**0.8085**

Table 3: Performances of runs on STS benchmark.

7 Conclusions

At SemEval 2017 STS task, we introduced a unsupervised knowledge based method, SIS, which could be new at SemEval. SIS is the extension of information content for STS evaluation. The performance of SIS is pretty good on STS test sets for it's just a new unsupervised method with room to improve. Currently, our main concern is how to gain the information contained in word embeddings, which may be lost in knowledge based SIS, and combine it with SIS to improve STS performance.

Acknowledgments

The work described in this paper is mainly supported by National Programs for Fundamental Research and Development of China (973 Program) under Grant 2013CB329303.

The authors would like to thank Daniel Cer and Eneko Agirre for their insightful comments to the improvement in technical contents and paper presentation.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo 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)*. Association for Computational Linguistics, pages 252–263. https://doi.org/10.18653/v1/S15-2045.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel

[4] http://ixa2.si.ehu.es/stswiki/index.php/STSbenchmark

Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 Task 10: Multilingual Semantic Textual Similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Association for Computational Linguistics, pages 81–91. https://doi.org/10.3115/v1/S14-2010.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, pages 497–511. https://doi.org/10.18653/v1/S16-1081.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Association for Computational Linguistics, pages 385–393. http://aclweb.org/anthology/S12-1051.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic Textual Similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*. Association for Computational Linguistics, pages 32–43. http://aclweb.org/anthology/S13-1004.

Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 2012. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Association for Computational Linguistics, pages 435–440. http://aclweb.org/anthology/S12-1059.

Steven Bird. 2006. NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*. Association for Computational Linguistics, pages 69–72. http://aclweb.org/anthology/P06-4018.

Tomáš Brychcín and Lukáš Svoboda. 2016. UWB at SemEval-2016 Task 1: Semantic Textual Similarity using Lexical, Syntactic, and Semantic Information. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, pages 588–594. https://doi.org/10.18653/v1/S16-1089.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, pages 1–14. http://www.aclweb.org/anthology/S17-2001.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2(3):27.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The weka data mining software: an update. *ACM SIGKDD explorations newsletter* 11(1):10–18.

Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. UMBC_EBIQUITY-CORE: Semantic Textual Similarity Systems. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*. Association for Computational Linguistics, pages 44–52. http://aclweb.org/anthology/S13-1005.

Aminul Islam and Diana Inkpen. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 2(2):10.

Paul Jaccard. 1908. *Nouvelles recherches sur la distribution florale*.

Jay J Jiang and David W Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference Research on Computational Linguistics (ROCLING X)*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Association for Computational Linguistics, pages 177–180. http://aclweb.org/anthology/P07-2045.

Yuhua Li, David McLean, Zuhair A Bandar, James D O'shea, and Keeley Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering* 18(8):1138–1150.

Dekang Lin. 1997. Using Syntactic Dependency as Local Context to Resolve Word Sense Ambiguity. In *35th Annual Meeting of the*

Association for Computational Linguistics. http://aclweb.org/anthology/P97-1009.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *ICML*. Citeseer, volume 98, pages 296–304.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, pages 55–60. https://doi.org/10.3115/v1/P14-5010.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*. volume 6, pages 775–780.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition ldc2011t07, 2011. *URL https://catalog. ldc. upenn. edu/LDC2011T07.[Online]* .

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. *International Joint Conference on Artificial Intelligence (IJCAI)* .

Barbara Rychalska, Katarzyna Pakulska, Krystyna Chodorowska, Wojciech Walczak, and Piotr Andruszkiewicz. 2016. Samsung poland nlp team at semeval-2016 task 1: Necessity for diversity; combining recursive autoencoders, wordnet and ensemble methods to measure semantic similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, pages 602–608. https://doi.org/10.18653/v1/S16-1091.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. Takelab: Systems for measuring semantic text similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Association for Computational Linguistics, pages 441–448. http://aclweb.org/anthology/S12-1060.

Shirish Krishnaj Shevade, S Sathiya Keerthi, Chiranjib Bhattacharyya, and Karaturi Radha Krishna Murthy. 2000. Improvements to the smo algorithm for svm regression. *IEEE transactions on neural networks* 11(5):1188–1193.

Arafat Md Sultan, Steven Bethard, and Tamara Sumner. 2014a. Back to Basics for Monolingual Alignment: Exploiting Word Similarity and Contextual Evidence. *Transactions of the Association of Computational Linguistics* 2:219–230. http://aclweb.org/anthology/Q14-1018.

Arafat Md Sultan, Steven Bethard, and Tamara Sumner. 2014b. DLS@CU: Sentence Similarity from Word Alignment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Association for Computational Linguistics, pages 241–246. https://doi.org/10.3115/v1/S14-2039.

Arafat Md Sultan, Steven Bethard, and Tamara Sumner. 2015. DLS@CU: Sentence Similarity from Word Alignment and Semantic Vector Composition. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, pages 148–153. https://doi.org/10.18653/v1/S15-2027.

Hao Wu and Heyan Huang. 2016. Sentence similarity computational model based on information content. *IEICE TRANSACTIONS on Information and Systems* 99(6):1645–1652.

Hao Wu and Heyan Huang. 2017. Efficient algorithm for sentence information content computing in semantic hierarchical network. *IEICE TRANSACTIONS on Information and Systems* 100(1):238–241.