L2F/INESC-ID at SemEval-2017 Tasks 1 and 2: Lexical and semantic features in word and textual similarity

Pedro Fialho, Hugo Rodrigues, Luísa Coheur and Paulo Quaresma
Spoken Language Systems Lab (L2F), INESC-ID
Rua Alves Redol 9
1000-029 Lisbon, Portugal
name.surname@l2f.inesc-id.pt

Abstract

This paper describes our approach to the SemEval-2017 "Semantic Textual Similarity" and "Multilingual Word Similarity" tasks. In the former, we test our approach in both English and Spanish, and use a linguistically-rich set of features. These move from lexical to semantic features. In particular, we try to take advantage of the recent Abstract Meaning Representation and SMATCH measure. Although without state of the art results, we introduce semantic structures in textual similarity and analyze their impact. Regarding word similarity, we target the English language and combine WordNet information with Word Embeddings. Without matching the best systems, our approach proved to be simple and effective.

1 Introduction

In this paper we present two systems that competed in SemEval-2017 tasks "Semantic Textual Similarity" and "Multilingual Word Similarity", using supervised and unsupervised techniques, respectively.

For the first task we used lexical features, as well as a semantic feature, based in the Abstract Meaning Representation (AMR) and in the SMATCH measure. AMR is a semantic formalism, structured as a graph (Banarescu et al., 2013). SMATCH is a metric for comparison of AMRs (Cai and Knight, 2013). To the best of our knowledge, these were not yet applied to Semantic Textual Similarity. In this paper we focus on the contribution of the SMATCH score as a semantic feature for Semantic Textual Similarity, relative to a model based on lexical clues only.

For word similarity, we test semantic equivalence functions based on WordNet (Miller, 1995) and Word Embeddings (Mikolov et al., 2013). Experiments are performed on test data provided in the SemEval-2017 tasks, and yielded competitive results, although outperformed by other approaches in the official ranking.

The document is organized as follows: in Section 2 we briefly discuss some related work; in Sections 3 and 4, we describe our systems regarding the "Semantic Textual Similarity" and "Multilingual Word Similarity" tasks, respectively. In Section 5 we present the main conclusions and point to future work.

2 Related work

The general architecture of our STS system is similar to that of Brychcín and Svoboda (2016), Potash et al. (2016) or Tian and Lan (2016), but we employ more lexical features and AMR semantics.

Brychcín and Svoboda (2016) model feature dependence in Support Vector Machines by using the product between pairs of features as new features, while we rely on neural networks. In Potash et al. (2016) it is concluded that feature based systems have better performance than structural learning with syntax trees. A fully-connected neural network is employed on hand engineered features and on an ensemble of predictions from feature based and structural based systems. We also employ a similar neural network on hand engineered features, but use semantic graphs to obtain one of such features.

For word similarity, our approach isolates the micro view approach seen in (Tian and Lan, 2016), where word embeddings are applied to measure the similarity of word pairs in an unsupervised manner. This work also describes supervised experiments on a macro/sentence view, which em-

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 213–219,
Vancouver, Canada, August 3 - 4, 2017.

ploy hand engineered features and the Gradient Boosting algorithm, as in our STS system.

Henry and Sands (2016) employ WordNet for their sentence and chunk similarity metric, as also occurs in our system for word similarity.

3 Task 1 - Semantic textual similarity

In this section we describe our participation in Task 1 of SemEval-2017 (Cer et al., 2017), aimed at assessing the ability of a system to quantify the semantic similarity between two sentences, using a continuous value from 0 to 5 where 5 means semantic equivalence. This task is defined for monolingual and cross-lingual pairs. We participated in the monolingual evaluation for English, and we also report results for Spanish, both with test sets composed by 250 pairs. Most of our lexical features are language independent, thus we use the same model.

For a pair of sentences, our system collects the numeric output of metrics that assess their similarity relative to lexical or semantic aspects. Such features are supplied to a machine learning algorithm to: a) build a model, using pairs labeled with an equivalence value (compliant with the task), or b) predict such value, using the model.

3.1 Features

In our system, the similarity between two sentences is represented by multiple continuous values, obtained from metrics designed to leverage lexical or semantic analysis on the comparison of sequences or structures. Lexical features are also applied to alternative views of the input text, such as character or metaphone[1] sequences. A total of 159 features was gathered, from which one relies on semantic representations.

Lexical features are obtained from INESC-ID@ASSIN (Fialho et al., 2016), such as TER, edit distance and 17 others. These are applied to 6 representations of an input pair, totaling 96 features since not all representations are valid on all metrics (for instance, TER is not applicable on character trigrams). Its metrics and input representations rely on linguistic phenomena, such as the BLEU score on metaphones of input sentences.

We also gather lexical features from HARRY[2], where 21 similarity metrics are calculated for bits, bytes and tokens of a pair of sentences, except for the Spectrum kernel on bits (as it is not a valid combination), resulting in 62 of our 159 features.

The only semantic feature is the SMATCH score (Cai and Knight, 2013) which represents the similarity among two AMR graphs (Banarescu et al., 2013). The AMR for each sentence in a pair is generated with JAMR[3], and then supplied to SMATCH, which returns a numeric value between 0 and 1 denoting their similarity.

In SMATCH, an AMR is translated into triples that represent variable instances, their relations, and global attributes such as the start node and literals. The final SMATCH score is the maximum F score of matching triples, according to various variable mappings, obtained by comparing their instance tokens. These are converted into lower case and then matched for exact equality.

3.2 Experimental setup

We applied all metrics to the train, test and trial examples of the SICK corpus (Marelli et al., 2014) and train and test examples from previous Semantic Textual Similarity in SemEval, as compiled by Tan et al. (2015).

Thus, our training dataset is comprised of 24623 vectors (with 9841 from SICK) assigned to a continuous value ranging from 0 to 5. Each vector contains our 159 feature values for the similarity among the sentences in an example pair.

We standardized the features by removing the mean and scaling to unit variance and norm. Then, machine learning algorithms were applied to the feature sets to train a model of our Semantic Textual Similarity representations. Namely, we employed ensemble learning by gradient boosting with decision trees, and feedforward neural networks (NN) with 1 and 2 fully connected hidden layers.

SMATCH is not available for Spanish, therefore this feature was left out when evaluating Spanish pairs (es-es). For English pairs (en-en), the scenarios include: a) only lexical features, or b) an ensemble with lexical features and the SMATCH score (without differentiation).

Gradient boosting was applied with the default configuration provided in scikit-learn (Pedregosa et al., 2011). NN were configured with single and multiple hidden layers, both with a rectifier as activation function. The first layer combines the 159

[1] Symbols representing how a word sounds, according to the Double Metaphone algorithm.

[2] http://www.mlsec.org/harry/

[3] https://github.com/jflanigan/jamr

input features (or 158 when not using SMATCH) into 270 neurons, which are either combined into a second layer with 100 neurons, or to the output layer (with 1 neuron). Finally, we employed the mean square error cost function and the ADAM optimizer (Kingma and Ba, 2014), and fit a model in 100 epochs and batches of 5.

Our experiments were run with Tensorflow 0.11 (Abadi et al., 2015), with NN implementations from the Keras framework[4]. Gradient boosting implementation is from scikit-learn.

3.3 Results

System performance in the Semantic Textual Similarity task was measured with the Pearson coefficient. A selection of results is shown in Table 1, featuring our different scenarios/configurations, our official scores (in bold), and systems that achieved results similar to ours or are the best of each language/track. Variations of our system are identified by the "*l2f*" prefix.

System	es-es	en-en
RTV (best of *en-en*)	0.6863	0.8547
ECNU (best of *es-es*)	0.8559	0.8518
neobility	0.7928	0.7927
l2f G. boost	0.7620	0.7919
l2f G. boost (+smatch)	-	**0.7811**
UdL	-	0.7805
MatrusriIndia	0.7614	0.7744
cosine baseline	0.71169	0.7278
l2f NN-1 (+smatch)	-	0.6998
l2f NN-1	0.6808	**0.6952**
l2f NN-2	0.6065	0.6832
l2f NN-2 (+smatch)	-	**0.6661**

Table 1: Pearson scores on monolingual evaluation, in descending order of performance on the English track.

We should mention that, afterwards, we ran our experiments with Theano 0.8.2, which yielded different results. As an example, on the English track, using the same settings (network topology, training data and normalization) of run "*l2f* NN-2 (+smatch)" resulted in a Pearson score of 0.72374. More recently, Tensorflow released version 1.0, which resulted in a score of 0.70437 for the same setup[5].

[4] https://keras.io/

[5] https://www.tensorflow.org/install/migration#numeric_differences

In order to evaluate the contribution of SMATCH, we analyzed some examples where SMATCH led to a lower deviation from the gold standard, and, at the same time, higher deviation from runs without SMATCH.

On 15 pairs, SMATCH based predictions were consistently closer to the gold standard, across all learning algorithms, with an average difference of 0.27 from non SMATCH predictions. However, after analyzing the resulting AMR of some of these cases, we noticed that information was lost during AMR conversion. For instance, consider the following examples, which led to the results presented in Table 2.

(A) *The player shoots the winning points. / The basketball player is about to score points for his team.*, with a gold score of 2.8.

(B) *A woman jumps and poses for the camera. / A woman poses for the camera.*, with a gold score of 4.0.

(C) *Small child playing with letter P / 2 young girls are sitting in front of a bookcase and 1 is reading a book.*, with a gold score of 0.8.

Considering example A, we can see the information lost during the AMR conversion in the following.

```
(w / win-01
 :ARG1 (p / point))
vs.
(b / basketball                    (1)
 :ARG1-of (s / score-01
 :ARG2 (t / team
 :location-of (p / point))))
```

The top structure (until "vs.") is the AMR for the first sentence, where "winning" is incorrectly identified as a verb, and the actual verb ("shoot") and its subject ("player") are missing. The same subject is also missing in the bottom AMR. For a comprehensive understanding of the AMR notation and the parser we employed please see Banarescu et al. (2013) and Flanigan et al. (2014), respectively.

The same happened with example B (and C, al-

	Algorithm	SMATCH	no SMATCH	Gold
A	G. boost	2.77	2.93	2.8
	NN-1	3.31	1.82	
	NN-2	2.00	1.74	
B	G. boost	4.13	4.15	4.0
	NN-1	4.00	4.31	
	NN-2	4.05	4.34	
C	G. boost	1.76	1.89	0.8
	NN-1	1.01	1.66	
	NN-2	1.32	1.89	

Table 2: Predictions for pairs A, B and C where SMATCH excels, grouped by pair.

though not presented here):

```
(a / and
 :op1 (p / pose-02
  :ARG0 (w / woman)
  :ARG1 (c / camera)))
vs.
(p / pose-02
 :ARG0 (w / woman)
 :location (c / camera))
```
(2)

Thus, we could not identify specific situations to which AMR explicitly contributed, since examples where using SMATCH yielded better results reveal that SMATCH was applied to AMR with less information than in the source sentence.

To conclude, we should say that 20 pairs were consistently better predicted without SMATCH, with an average difference to SMATCH based predictions of 0.38.

4 Task 2.1 - Multilingual word similarity: English

In this section we report the experiments conducted for the second task of 2017 SemEval (Camacho-Collados et al., 2017). The task consists of, given a pair of words, automatically measuring their semantic similarity, in a continuous range of $[0-4]$, from unrelated to totally similar. The test set was composed of 500 pairs of tokens (which can be words or multiple-word expressions); a small trial of 18 pairs set was also provided by the organizers.

For this task we used a family of equivalence functions, from now on $equiv(t_1, t_2)$, where t_1 and t_2 are the tokens to be compared. $equiv$ functions return a value in the range $[0-1]$. This value was later scaled into the goal's range. Then, we analyzed how to combine them. In the following subsections we detail our approach.

4.1 Equivalence functions

Two functions were considered:

- $equiv_{WN}$, which uses WordNet (Miller, 1995).
- $equiv_{W2V}$, which employs Word2Vec vectors (Mikolov et al., 2013) to compare the two tokens – we use the pre-trained vectors model available, trained on the Google News dataset[6].

$equiv_{WN}(t_1, t_2)$ is defined as:

$$equiv_{WN} = \begin{cases} 1 & \text{if } syn(t_1) = syn(t_2) \\ x & \text{if } syn\big(hyp(t_1)\big) \supset hyp(t_2) \\ x & \text{if } hyp(t_1) \subset syn\big(hyp(t_2)\big) \\ 0 & \text{otherwise,} \end{cases}$$

where:

- $syn(t)$ gives the synset of the token t;
- $hyp(t)$ gives the hypernyms of t;
- $x = 1 - max(n \times 0.1, m \times 0.1)$, with n and m being the number of nodes traversed in the synsets of t_1 and t_2, respectively.

$equiv_{WN}$ matches, thus, two tokens if they have a common hypernym (Resnik, 1995) in their synset path. We compute the path distance by traversing the synsets upwards until finding the least common hypernym. For each node up, a decrement of 0.1 is awarded, starting at 1.0. If, no concrete common hypernym is found, then 0 is the result returned.

Token 1	Token 2	$equiv_{WN}(\times 4)$	$equiv_{W2V}(\times 4)$	Gold
eagle	falcon	0.8 (3.20)	0.44 (1.76)	3.72
keyboard	light	0.7 (2.80)	0.02 (0.09)	0.24
science fiction	comedy	0.0 (0.00)	0.34 (1.37)	2.78
sunset	string	0.0 (0.00)	0.09 (0.36)	0.05

Table 3: Results of our functions in some instances of the trial set.

TeamName	Pearson	Spearman	Final
Luminoso_run2	0.783	0.795	0.789
Luminoso_run1	0.781	0.794	0.788
QLUT_run1	0.775	0.781	0.778
hhu_run1	0.71	0.699	0.704
HCCL*_run1	0.675	0.7	0.687
	...		
l2f(a.d.)_run2	0.644	0.654	0.649
l2f(a.d.)_run1	0.637	0.648	0.643
	...		
SEW_run1	0.373	0.414	0.392
hjpwhuer_run1	-0.037	-0.032	0.0

Table 4: Results for the runs submitted for Task 2.1 - English.

For example, *laptop* and *notebook* have the common synset `Portable Computer`, one node above both words, which results in a score of $1 - 0.1 = 0.9$. *Crocodile* and *lizard* return 0.8, as one needs to go up two nodes in both tokens to find the common synset `Diapsid`. We do not consider generic synsets such as `artifact` or `item`.

Regarding $equiv_{W2V}$, it computes the cosine similarity between the vectors representing the two tokens:

$$equiv_{W2V}(t_1, t_2) = \cos\big(W2V(t_1), W2V(t_2)\big),$$

where $W2V(t)$ is the vector representing the word embedding for the token t. If the token is composed by more than one word, their vectors are added before computing the cosine similarity. For example, *self-driving car* and *autonomous car* obtain a cosine similarity of 0.53 (showing a degree of similarity, resulting from multiple-word tokens), while *brainstorming* and *telescope* result in a score of 0.04, which means the tokens are not related. Note that the scores are rounded to 0 if they are negative.

[6] https://code.google.com/archive/p/word2vec/

4.2 Combining the equivalence functions

We started by applying the $equiv(t_1, t_2)$ to the trial set. Table 3 shows some results for this experience. As one can see, in certain cases it would be better to use $equiv_{WN}$, and in others the $equiv_{W2V}$ function.

Just these few examples show how hard it is to combine these functions. Although we did not expect to accomplish relevant results with such approach, we decided to train a linear regression model in Weka (Hall et al., 2009) with the (very small) provided example set.

The final result obtained was $C_1 = 5.0381 \times equiv_{w2v} + 0.6355$, which only uses one of the functions. We used this equation in one of our runs, `RunW2V`, with a modified version: $C' = min(C_1, 4)$.

Believing $equiv_{WN}$ had potential to be important in certain cases, we manually designed a weighed function to combine both functions. The threshold was decided by analyzing the trial set only. We ended up with the following decision function:

$$C_2 = \begin{cases} equiv_{WN} \times 4 & \text{if } equiv_{W2V} < 0.12 \\ C' & \text{otherwise.} \end{cases}$$

The idea behind it is the following: when $equiv_{W2V}$ is below a threshold (set to 0.12), we use $equiv_{WN}$. Then either $equiv_{WN}$ does not find a relation as well (and probably has a value of 0.0), or it finds one and it is probably correct (see *sunset/string* in Table 3). This led to our second run, `RunMix`.

4.3 Results

Results for the task are presented in Table 4, with our runs in bold as submitted (`run1` is `RunW2V` and `run2` is `RunMix`). Both our runs attain a similar score, which is somehow surprising given how differently the scores were calculated. We placed at the middle of the table, although only a few points short from the 5th best ranked run - a difference of less than 0.04 on both Pearson and final score. This ends up being an interesting result, based on how simple our approach was, and the lack of data to properly learn a function to combine our $equiv$ functions.

5 Conclusions and Future Work

In this paper we present our results on two tasks of 2017 SemEval competition, "Semantic Textual Similarity" and "Multilingual Word Similarity". The results obtained yielded competitive results, although being outperformed by other approaches in the official ranking.

For the "Semantic Textual Similarity" task, our models performed similarly for multilingual data, since most features are language independent, and essentially rely on matching tokens among input sentences. Therefore, our method is feasible for all monolingual pairs.

We could not identify situations where the SMATCH metric improved the results, although in 15 cases SMATCH based predictions were closer to the gold standard, across all learning algorithms.

Future work includes replacing the exact instance matching in SMATCH with our word similarity module, and using the SMATCH representation in a structural learning method such as Tree-LSTM (Tai et al., 2015), or in a more balanced/weighed ensemble with the lexical features.

In what respects the "Multilingual Word Similarity" task, we believe that our participation was simple, but still effective. We used two semantic resources (WordNet and Word2Vec), a weighting function learned on a small trial set, and a hand-crafted formula to combine the similarity scores of our two functions, which makes it an approach lacking ground. The results were still promising, given the simplicity of our approach.

As future work, the word similarity module itself could be largely improved by automatically learning a set of weights to combine the two functions. For instance, the gold standard, now available, can be a useful tool for this task, as other large datasets like Simlex-999 (Hill et al., 2014).

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. http://tensorflow.org/.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. Association for Computational Linguistics, Sofia, Bulgaria, pages 178–186. http://www.aclweb.org/anthology/W13-2322.

Steven Bethard, Daniel M. Cer, Marine Carpuat, David Jurgens, Preslav Nakov, and Torsten Zesch, editors. 2016. *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*. The Association for Computer Linguistics. http://aclweb.org/anthology/S/S16/.

Tomás Brychcín and Lukás Svoboda. 2016. UWB at semeval-2016 task 1: Semantic textual similarity using lexical, syntactic, and semantic information. In (Bethard et al., 2016), pages 588–594. http://aclweb.org/anthology/S/S16/S16-1089.pdf.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*. The Association for Computer Linguistics, pages 748–752.

Jose Camacho-Collados, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli. 2017. Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, pages 15–26. http://www.aclweb.org/anthology/S17-2002.

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.

Pedro Fialho, Ricardo Marques, Bruno Martins, Luísa Coheur, and Paulo Quaresma. 2016. Inesc-id@assin: Medição de similaridade semântica e reconhecimento de inferência textual. *Linguamática* 8(2):33–42.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, volume 1, pages 1426–1436.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: An update. *SIGKDD Explor. Newsl.* 11(1):10–18. https://doi.org/10.1145/1656274.1656278.

Sam Henry and Allison Sands. 2016. Vrep at semeval-2016 task 1 and task 2: A system for interpretable semantic similarity. In (Bethard et al., 2016), pages 577–583. http://aclweb.org/anthology/S/S16/S16-1087.pdf.

Felix Hill, Roi Reichart, and Anna Korhonen. 2014. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *CoRR* abs/1408.3456. http://arxiv.org/abs/1408.3456.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980. http://arxiv.org/abs/1412.6980.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*. European Language Resources Association (ELRA), pages 216–223. http://www.lrec-conf.org/proceedings/lrec2014/summaries/363.html.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

George A. Miller. 1995. Wordnet: a lexical database for english. *Commun. ACM* 38:39–41.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Peter Potash, William Boag, Alexey Romanov, Vasili Ramanishka, and Anna Rumshisky. 2016. Simihawk at semeval-2016 task 1: A deep ensemble system for semantic textual similarity. In (Bethard et al., 2016), pages 741–748. http://aclweb.org/anthology/S/S16/S16-1115.pdf.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, IJCAI'95, pages 448–453.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. The Association for Computer Linguistics, pages 1556–1566. http://aclweb.org/anthology/P/P15/P15-1150.pdf.

Li Ling Tan, Carolina Scarton, Lucia Specia, and Josef van Genabith. 2015. Usaar-sheffield: Semantic textual similarity with deep regression and machine translation evaluation metrics. In *Proceedings of the 9th International Workshop on Semantic Evaluation*. o.A., pages 85–89.

Junfeng Tian and Man Lan. 2016. ECNU at semeval-2016 task 1: Leveraging word embedding from macro and micro views to boost performance for semantic textual similarity. In (Bethard et al., 2016), pages 621–627.