LIPN-IIMAS at SemEval-2017 Task 1: Subword Embeddings, Attention Recurrent Neural Networks and Cross Word Alignment for Semantic Textual Similarity

Ignacio Arroyo-Fernández
Universidad Nacional
Autonoma de Mexico, Mexico
iaf@ciencias.unam.mx

Ivan Meza
Instituto de Investigaciones
en Matematicas Aplicadas y en Sistemas
Universidad Nacional
Autonoma de Mexico, Mexico
ivanvladimir@turing.iimas.unam.mx

Abstract

In this paper we report our attempt to use, on the one hand, state-of-the-art neural approaches that are proposed to measure Semantic Textual Similarity (STS). On the other hand, we propose an unsupervised cross-word alignment approach, which is linguistically motivated. The neural approaches proposed herein are divided into two main stages. The first stage deals with constructing neural word embeddings, the components of sentence embeddings. The second stage deals with constructing a semantic similarity function relating pairs of sentence embeddings. Unfortunately our competition results were poor in all tracks, therefore we concentrated our research to improve them for Track 5 (EN-EN).

1 Introduction

Semantic Textual Similarity (STS) refers to the Natural Language Processing (NLP) task which is aimed at measuring the degree of similarity/dissimilarity between two text units (Agirre et al., 2012, 2016). In other words given a pair of text snippets (generally a pair of sentences) the task is to determine a real value (the semantic similarity score) in the interval between 0.0 and 5.0, which represents how much similar are the two sentences of a given pair.

There are two main types of proposed systems in prior editions of the competition: supervised and unsupervised systems. While supervised systems are expected to be highly reliable because of that they use human-annotated gold standards, unsupervised systems also are highly reliable by using modest levels of linguistic knowledge. In this work we report results from both, unsupervised and supervised systems.

Currently the STS task involves tracks of different nature, i.e. the monolingual and cross-lingual ones. In this paper we investigate the underlying properties in text which are relevant to measure semantic similarity, thus we focus our major efforts into the English-English Track 5.

2 Data

We tested a couple of supervised systems. We prepared the STS monolingual English datasets from years 2012, 2013, 2015 and 2016. After discarding sentence pairs whose similarity score was absent from the corresponding gold standard files, we obtained a dataset consisted of $10,592$ sentence pairs ($6,858$ are already marked as training pairs and $3,734$ are already marked as test pairs).

In order to obtain subword embeddings we trained the *“fastText”* method for 20, 50, 100, 200 and 300 dimensions by using the English Wikipedia (Bojanowski et al., 2016). We decided to take advantage of the capability of this method for inferring out-of-vocabulary words. This advantage is mainly due to the fastText’s character level n-gram approach, which represents a meaningful performance difference both in training and in testing.

3 Systems Description

Multiple Neural Network architectures were used to model similarity measuring in supervised settings. Also an unsupervised system[1] was directly tested on this year’s evaluation dataset.

3.1 Word embeddings + RNN

We see the Recurrent Neural Networks (RNN) as intuitive models for observing relevance of sentence elements; in particular the Long-Short Term Memories (LSTMs). These kind of networks are

[1] https://github.com/iarroyof/sts_align

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 208–212,
Vancouver, Canada, August 3 - 4, 2017. ©2017 Association for Computational Linguistics

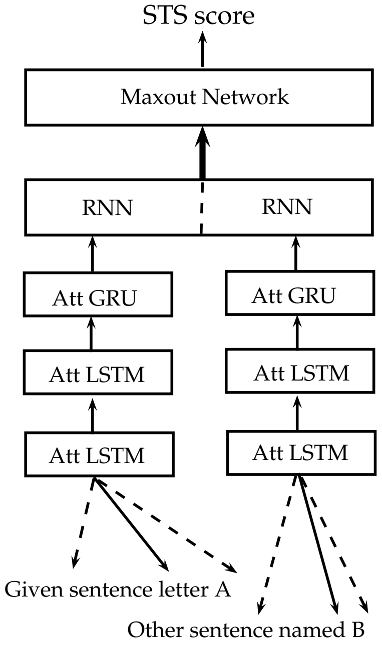

Figure 1: Attentional architecture for detecting relevant parts of each sentence within a pair of sentences.

well documented as suitable for modeling sequentiality of lexical units within sentences whereas avoiding the gradient vanishing of long term patterns (Hochreiter and Schmidhuber, 1997).

In the case of Attention LSTMs, they capture additional features of the sequential process they model. The additional features are encoded into an attention vector. This attention vector indicates to the network which segments of the sequence (sentence) are statistically more relevant than the other ones according to the training set.

In this paper we used the architecture proposed by (Vinyals et al., 2015), where the authors used a stacked Attention LSTM for PoS tagging. In Figure 1 we show a modified version of the mentioned architecture, which consists of two attention LSTM layers on the bottom, one Gated Recurrent Unit (GRU) at the middle and a simple RNN on top (Cho et al., 2014). Notice that this description corresponds to each of the twin networks showed in the figure, which is our adaptation to the STS task. This recurrent architecture is followed by a Maxout Network (Goodfellow et al., 2013), which has a monolithic output layer (i.e. the similarity score $y_i \in [1, 5] \subset \mathbb{R}$).

3.2 Sentence embeddings + MLP

The word/sentence embedding stage was modeled via the *doc2vec* method (Le and Mikolov, 2014), which is based on the *word2vec* word embedding method (Mikolov et al., 2013). For each pair of sentences, we obtained a pair of sentence embeddings $(s_a, s_b) \in \mathbb{R}^d \times \mathbb{R}^d$. Thus each pair was concatenated to form a pair vector $p_i = s_a \| s_b \in \mathbb{R}^{2d}$. In this way, we obtained a training set $(p_1, y_1), ..., (p_m, y_m)$ which was feed to a simple MLP. The output layer of the MLP is a 6-node softmax, so we have six possible output similarity values, i.e. $y_i \in \{0, ..., 5\}$.

3.3 Cross word aligner

We proposed an unsupervised system which is motivated by linguistic elements we identified as highly relevant accordingly to linguistic theories. General linguistics states that we can know what is being said about something by seeing at the predicative structure. The theories by Harris (1968) inspire NLP algorithms where it is said that word use leads to meaning (which is commonly interpreted as word co-occurrence). Harris also said that combinatorics of words is more informative in the predicates, where redundancy is needed by speakers to provide integrity to a message.

In an attempt to follow these statements and also inspired by success obtained by authors like Han et al. (2013) and Rychalska et al. (2016), we implemented a word alignment system. Unlike previous works, our system considers that verbs operate on nouns. We used Open Information Extraction algorithms (openIE) for detecting predicates $(\mathcal{P}_a, \mathcal{P}_b)$ of the form (NP, VP, NP) within each sentence of the pair (S_a, S_b) (Fader et al., 2011).

Similarly to the word analogies commonly used for word embedding evaluations (Mikolov et al., 2013), our system considers that verbs frequently operate on nouns. Thus, it is measured how similar each verb $v_a \in \mathcal{P}_a$ of a sentence S_a is, with respect to its combination with each noun $n_b \in \mathcal{P}_b$ of a sentence S_b, i.e. $d_c(S_a, S_b)$. Given that the relationship $d_c(\cdot, \cdot)$ is not commutative this similarity also is computed from S_b to S_a, i.e.

$$d_c(S_a, S_b) = \frac{1}{N_{v,a}} \sum_{v_a \in S_a} \frac{1}{N_{n,b}} \sum_{n_b \in S_b} \theta(v_a, n_b) \tag{1a}$$

$$d_c(S_b, S_a) = \frac{1}{N_{v,b}} \sum_{v_b \in S_b} \frac{1}{N_{n,a}} \sum_{n_a \in S_a} \theta(v_b, n_a), \tag{1b}$$

where $\theta(\cdot, \cdot)$ is the cosine similarity and $v_a, n_a \in \mathbb{R}^d$ are word embeddings categorized as verbs

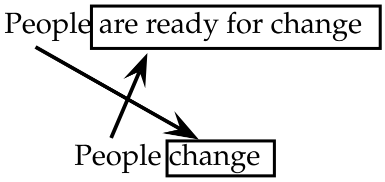

Figure 2: General scheme for the vector similarities of cross word alignments with respect to structural categories.

and nouns within the sentence S_a, respectively. $N_{v,a}, N_{n,a}$ are the number of verbs and nouns considered in S_a (same for S_b). Overall, equations (1a) and (1b) are the average vector similarities of *cross word alignments* with respect to structural categories between S_a and S_b. For example, in Figure 2 the sentence "People are ready for change" $[S_a]$ is compared against the phrase "people change" $[S_b]$. The main idea, in one direction $[S_b] \rightarrow [S_a]$, is to quantify how the word "people" is used along the conjugated form "are" (which forms a predicate together with the noun phrase "ready for change"). This operation is also performed in the inverted direction $[S_a] \rightarrow [S_b]$.

The kind of predicates showed in Figure 2 are often part of more complex sentences, e.g. "It is clear that future is near and people is ready for change". We extracted these predicates by using the openIE algorithm implemented in the coreNLP[2] library.

There are cases in the STS corpora where no extractions are made. This is due to the low recall openIE systems offer until now (Xu et al., 2013). That is, many openIE algorithms can extract neither implicit relations (e.g. "Mexico City, where Aztecs live") nor short phrases (e.g. "The white house"). We assume that these snippets are expressed in their minimum form, so things like "people changes" are embedded word by word. The embeddings are then compared either to embeddings of other equally short phrases or to embeddings of openIE extractions. The global score is simply the average of all distances:

$$s_f = \frac{d_c(S_a, S_b) + d_c(S_b, S_a)}{2}$$

4 Results

Our systems passed through several refinement stages. Unfortunately, the submitted runs were to early stages and did not reach competitive performance as can be seen in Table 1. We transformed the multi-lingual data onto English using the Google Translate API and trained a unique model on resulting data. We submitted two LSTM models, with and without attention mechanism. The models were selected by monitoring the best test score after 25 training epochs. Additional systems were tested after-competition. Our best results are considered as such given its absolute value (inverse correlations can be reinterpreted in-system in the case we reach higher values).

	LSTM	Attention
Track 1	0.0471	0.0214
Track 2	0.0769	0.1292
Track 3	0.1527	0.0458
Track 4	0.1719	0.0120
Track 5	0.1446	0.0191
Track 6	0.0738	0.2038
Track 7	0.0800	0.2168
Overall	0.1067	0.0926

Table 1: LSTM network without/with attention mechanism. Official results of the competition in this year's evaluation.

4.1 Word embeddings + RNN

A sentence can be seen as a sequence of word embeddings which are appended in order to form a sentence matrix. For this system we used FastText word embeddings. Given a sentence pair, each sentence matrix is fed to each of the multi-layered RNNs described in Section 3.1. We used the last-top hidden states (or time steps) of the two networks as sentence embeddings. We concatenated these sentence embeddings. In this way, we obtained pair vectors $p_1, ..., p_m \subset \mathbb{R}^{2t}$ that were feed to the top Maxout network (herein t is the number of hidden states each of the top RNN layers has in Figure 1).

The networks showed in Table 2 were trained over 1500 pairs from data described in Section 2 (1050 for training and 450 for test). As shown in the table, we fed the networks with word embeddings of 200, 100 and 50 dimensions. Results are much better for the architecture formed by word embeddings of 200 dimensions, 50 hidden states and 100 hidden Maxout nodes.

[2] http://stanfordnlp.github.io/CoreNLP/

Time steps	Hidden Maxout	Embedding dimension	Correlation/ MSE
50	100	200	-0.2951/1.2
100	40	100	-0.2848/1.8567
25	10	50	-0.0103/2.0738
10	40	50	-0.0123/2.0252

Table 2: Twin Attention LSTM-GRU-RNN-Maxout architecture and performance (**after-official evaluation**) on the 2017 track 5.

4.2 Sentence embeddings + MLP

We trained Doc2vec sentence embeddings $s_a, s_b \in \mathbb{R}^d$ of different dimensions (i.e. d=100, 200, 300, 500, 600) by using the whole data described in Section 2. All values of d other than 300 showed very poor learning in the MLP stage. Thus, we reported only results produced by 300-dimensional sentence embeddings.

Hidden layers	MSE (%)	Correlation
[210, 45]	64.56	0.0777
[260, 66]	64.67	0.0349
[250, 75]	64.94	0.0140
[80]	62.95	-0.0058
[270, 60]	65.32	0.0139

Table 3: Multilayer Perceptron architecture and performance in this year's evaluation (track 5).

In Table 3 we depict the Mean Squared Error (MSE) for the test set and the Pearson's weighted correlation coefficient for the track 5 evaluation. Many combinations in the architecture during the training showed that even the minimum test MSE is very high. Therefore our setting Doc2vec+MLP did not allow for good generalization.

4.3 Cross word aligner

The cross word alignment system is unsupervised and we tested it directly on some of the most popular past year's datasets. We used fastText word embeddings of different dimensions. A good choice for semantic assessment is 100 dimensions (Bojanowski et al., 2016). Additionally we reported results for 300, 200, 50 and 20 dimensions.

On top of Table 4 we show our best result (after official evaluation), which is that for 200 dimensions. Furthermore we noticed our engineered features are sensitive to text properties, e.g. domains and, therefore, writing styles are very different between Headlines and Eval. 2017. It is needed to say that we tested direct word alignments (i.e. verb-verb, noun-noun) without success.

Corpus	Dim.	Correlation
Eval. 2017	200	-0.4599
Eval. 2017	100	-0.4557
Eval. 2017	50	-0.4291
Eval. 2017	20	-0.3716
Eval. 2017	300	-0.3597
OnWN	200	-0.4389
Plagiarism	100	-0.1851
Headlines	20	-0.1481

Table 4: Cross word aligner results. This year's evaluation and best results for popular STS data.

5 Conclusions

Despite of the success that RNNs have recently showed, we observed that even when they do not require feature engineering, instead they require training time, large data amounts, high computational power and architecture engineering. The results we showed in Section 4.1 are not good. The reason is very probably one the aforementioned and it needs to be improved. We think the amount of sequential patterns with which we trained our networks was not enough. Such patterns are based on punctual lexical items (each particular word embedding), but not in generalized sequential and semantic patterns.

Our cross word alignment system is based on feature engineering, in such a way that we showed that when a simple cosine similarity focuses on relevant segments of sentences, the performance can be progressively improved (probably by improving feature engineering and adding external resources not considered at this moment). This reasoning is consistent with much other unsupervised approaches. It is needed to say that even when we performed simple feature engineering, a critical part of our method was the use of word embeddings, which are barely based on linguistic feature engineering.

Acknowledgments

We thank to *Laboratorio Universitario de Cómputo de Alto Rendimiento* (IIMAS-UNAM); to The CONACyT (grant No. 386128) and to the CS graduate program (UNAM).

References

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. *Proceedings of SemEval* pages 497–511.

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of 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*. Association for Computational Linguistics, pages 385–393.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606* .

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* .

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1535–1545.

Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron C Courville, and Yoshua Bengio. 2013. Maxout networks. *ICML (3)* 28:1319–1327.

Lushan Han, Abhay Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. Umbc ebiquity-core: Semantic textual similarity systems. *Atlanta, Georgia, USA* 44.

Zellig S. Harris. 1968. *Mathematical Structures of Language*. Wiley, New York, NY, USA.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*. pages 1188–1196. http://jmlr.org/proceedings/papers/v32/le14.html.

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

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), San Diego, CA, USA*.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*. pages 2773–2781.

Ying Xu, Mi-Young Kim, Kevin Quinn, Randy Goebel, and Denilson Barbosa. 2013. Open information extraction with tree kernels. In *HLT-NAACL*. pages 868–877.