Measuring Semantic Similarity Between Sentences Using a Siamese Neural Network

Alexandre Yukio Ichida
Felipe Meneguzzi
Duncan D. Ruiz
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, Brazil
alexandre.ichida@acad.pucrs.br, {felipe.meneguzzi,duncan.ruiz}@pucrs.br

***Abstract*—The task of measure semantic redundancy between sentences demands a thorough interpretation from the reader because phrase meaning may be ambiguous. Detecting semantic similarity is a difficult problem because natural language, besides ambiguity, offers almost infinite possibilities to express the same idea. This paper adapts a siamese neural network architecture trained to measure the semantic similarity between two sentences through metric learning. The resulting solution should help in writing more efficient and informative text.**

***Index Terms*—Neural networks, word embedding, recurrent neural network, GRU, metric learning, siamese neural networks, semantic analysis**

I. Introduction

Semantic similarity is a quantitative measure that shows closeness of meaning given different pieces of text, regardless of how they are written. The key challenge in comparing the semantic content of natural language lies in the very large number of different ways of expressing the same information, especially when reasoning about the context surrounding the text [1]. Computing semantic similarity allows us to objectively assess text passages in the same document for redundancy, helping writers convey the same information using less repetition.

Text redundancy occurs when the same document contains multiple passages of information with a high semantic similarity, expound the same idea in different areas of a text. Although redundancy can be used to emphasize some conclusion, it may generate unnecessary textual volume which results in a vague and uninteresting text.

We provide two contributions in this paper. First, we show that semantic similarity can be measured through learning a metric using a Siamese GRU (Gated Recurrent Unit) network architecture (Section III), which is trained using a labeled dataset (Section IV). Second, given a representation that encodes semantic and syntactic information about the words, we show that our approach to measure semantic similarity does not depend on linguistic information of the sentences.

II. Background

Neural network is a computational model used for machine learning purposes described as a direct acyclic graph [2], which is organized in multiple layers. Each intermediate layer, also known as hidden layer, can learn different representations given a input data. Recently, natural language process systems are applying neural networks to learn text representations, using techniques such as Word Embedding to create a word vector which reflects semantic syntactic properties of words. [3].

Taking into a sentence level, a classical feed forward neural network is limited to process each word of sentence as a single feature [2], ignoring their order of occurrence. Recurrent neural network is a neural network type for sequential data process, making possible learn information from the context of words considering their previous information in the sentence. Although the weights is shared accross the sequence using the same updating rule, training a recurrent neural network is difficult because gradients may become small over long sequences, being susceptible to vanishing/exploding gradient problem [4].

GRU [5] (Gated Recurrent Units) is a recurrent network architecture proposed to deal with long sequences, using a gating mechanism to create a memory control of values processed over time. A GRU cell consists of two gates that controls flow data through states. Gate r_t controls updates on internal memory, which is not propagated to next state. Gate z_t controls how much of internal memory should be considered on next state. Equations 1, 2 and 3 represents operations realized by gates r_t and z_t in order to results , and equation 3 shows how next hidden state is computed in a GRU unit [2].

$$r_t = \sigma(W_r h_{t-1} + U_r x_t + b_r) \tag{1}$$

$$z_t = \sigma(W_z h_{t-1} + U_z x_t + b_z) \tag{2}$$

$$h_t = z_t \otimes h_{t-1} \oplus (1-z) \otimes tanh(W_h x_t + U_h(r_t \otimes h_{t-1}) + b_h)) \tag{3}$$

In order to measure similarities, metric learning is an alternative to learn a distance function in a supervised manner, which considers changes in the data [6]. Siamese neural network is a architecture composed by two neural networks that is used to do metric learning between the output of each network. Chopra, Hadsell, and LeCun proposed [7] a Siamese neural network architecture to learn a distance function for face verification, which uses two symmetric convolutional networks. Mueller and Thyagarajan work [8] shows that learning a simple Manhattan Distance function over semantic encoding

of sentences can efficiently measure semantic similarity. We propose a similar way to measure semantic similarity, using a Siamese neural network architecture using two GRU (Gating Recurrent Units), which is a simpler architecture than a Long-Short Term Memory network used in Mueller and Thyagarajan work [8].

III. Model Architecture

In this section, we describe in two steps the architecture of our neural network used to obtain the semantic similarity between two sentences. First, we describe the data pre-processing in order to create a numerical representation to the words in sentences. Second, we detail the architecture of our neural network specifying the layers and explaining their roles and motivations of each selected approach.

A. Data Pre-processing

Since the input data of a neural network must be numeric values, the pre-processing step consists of creating a numerical representation to a sentence by converting each word into an integer number. For such conversion, we create a word dictionary of the corpus vocabulary and associate a unique numerical index for each word seen in the dataset. Thus, the dataset is entirely read before creating the numerical representation to sentences, in order to recognize all the words contained in the dataset.

The idea behind the word dictionary is to create a vector of integers for each sentence, being composed of word indexes. We maintain word order from sentences in the resulting vector, preserving the original context and meaning of the sentence semantics. Thus, our model can distinguish sentences which have same words in different position such sentence pair "a big dog in a small house" and "a small dog in a big house".

In order to prevent the same terms being associated with different indexes, we convert abbreviations contained in the dataset before inserting into the dictionary. For example, "what's" is converted to "what is" resulting on the use of index of the words "what" and "is", which may already seen in different contexts. This conversion allows us to reuse indexes already seen, reducing the size of the dictionary structure.

B. Siamese GRU Model

We use a Siamese architecture based on Mueller *et al* [8] due to its notable results on predicting the semantic similarity between sentences. Our modified Siamese neural network uses two symmetric recurrent neural networks with shared weights to learn semantic differences between sentences.

The input layer of our architecture converts each vector of indexes received from data pre-processing into a word distributed representation. Due to its capability on capture semantic and syntactic properties of the words in the result representation [3], we use *Word2Vec Skip-Gram* model pre-trained on an external corpus. Thus, our approach does not depend on a manual feature extraction process to represent input words with efficiently.

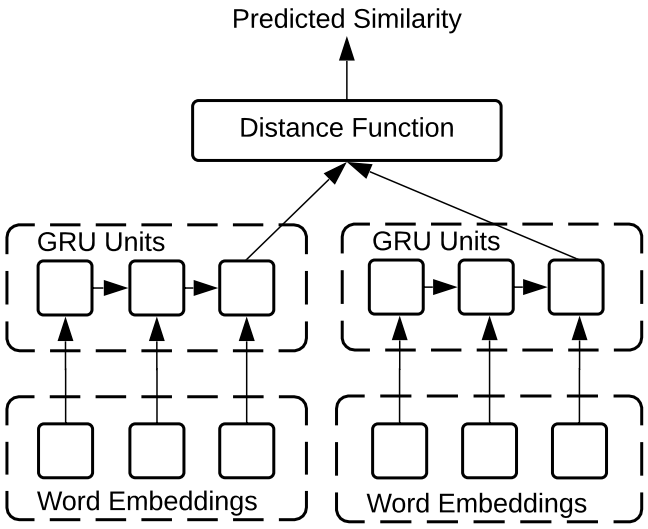

Fig. 1. Diagram ilustrating our siamese neural network defined, combining two recurrent neural network. The siamese network receives a sequence of word embeddings of the words contained in the input sentences, encodes into a sentence vector and learns a distance function to measure similarity.

In order to generate a sentence representation based on the word embedding, we make use of recurrent network architecture to process the word embedding sequences. Represented by a sequence of word vectors, we pass each sentence to recurrent units that update the hidden unit h_t of each state and learn to encode the entire sentence. To address the vanishing gradient problem [9], which a recurrent network is subjected when processing long sequences, we use the GRU architecture to control the gradient updates in the training process. Figure 2 shows the process of encoding the sequence of word embedding receives from previous layer.

The Siamese recurrent network in Mueller *et al* relies on an

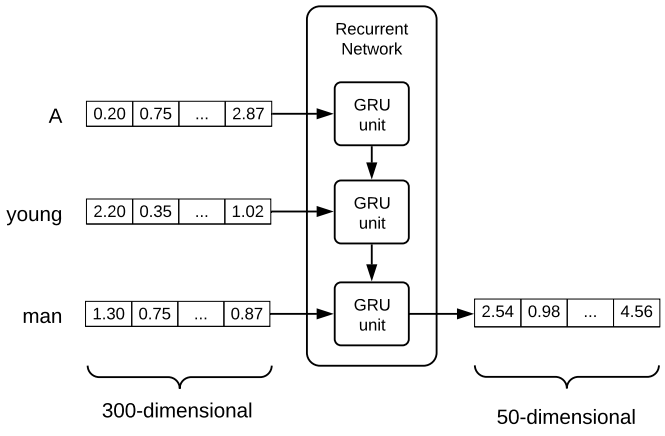

Fig. 2. Diagram ilustrating our defined recurrent neural network to encode sentences. The word vectors of sentence "a young man" is encoded to a 50-dimensional vector.

LSTM architecture [10] to create a sequence encoder mapping for each sentence. In order to overcome the limited size of the labeled datasets available to us we use a GRU architecture since GRU units have fewer parameters than LSTM units. To the best of our knowledge, GRU is less explored than LSTM on the context of semantic representation, even it has not been proved a general superiority of LSTM [11].

The output layer of the siamese network learns a distance function which results in a similarity metric between two encoded sentences. Our distance function used for the metric learning follows the work proposed by Mueller *et al* that uses the Manhattan Distance function to calculate the difference between the encoded sentences. Equation 4 shows the distance function used to measure the similarity in the output layer, where h_a and h_b represent the outputs of each recurrent network.

$$\exp(-||h_a - h_b||_1) \tag{4}$$

The Equation 4 is based on measure similarity between two representation, applying the exponential function on the negative value of distance measured by Manhattan Distance function. Due to the use of exponential function on negative numbers, the siamese network predicted value is a float number $y_t \in [0, 1]$. Thus, the error propagated during training underlies only the similarity predicted and the label value of the pair of sentence.

IV. Implementation and Training the Model

In this section, we detail about the implementation and execution of the neural network. First, we describe the data set used and its motivations to train the model. Second, we describe the parameters used for the training executions, initialization of weights and number of units for each layer.

A. SICK Dataset

The SICK data set (Sentences Involving Compositional Knowledge) [1] is provided by SemEval-2014[1] for predicting the degree of relatedness between sentences and detecting the entailment relation between them. The data set contains 10000 English sentence pairs extracted from the ImageFlick dataset[2] and SemEval-2012 semantic textual similarity video description data set. The use of SICK data set is motivated by the fact of possibility to train our model in a supervised manner, besides this dataset is used in related works, being suitable for comparisons.

The relatedness value annotated for each sentence pair is a numeric value between 1 and 5, representing the degree of semantic similarity between the two sentences. In this work, we do not use the entailment annotation available to this task since it is not our goal. This dataset is divided into three parts: 5000 sentence pairs as training set, 500 as validation set, and 4500 as test set.

[1]http://alt.qcri.org/semeval2014/

[2]http://nlp.cs.illinois.edu/HockenmaierGroup/data.html

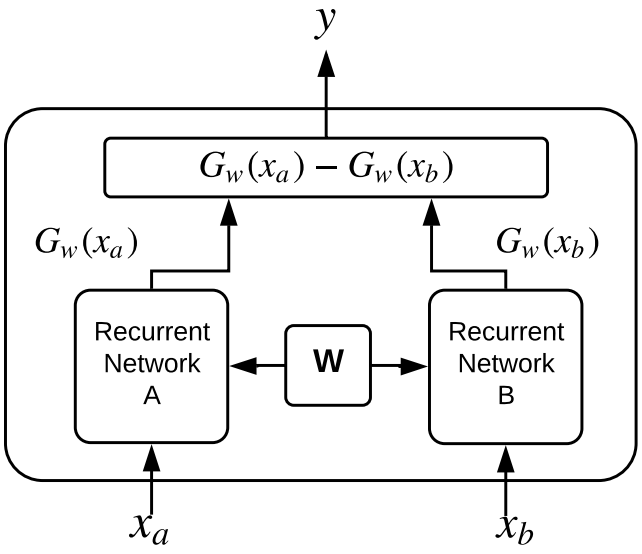

Fig. 3. Diagram ilustrating weight structure W shared between two networks, predicting y through a metric learning given two inputs x_a and x_b, which is applied a G_w function.

B. Implementation

We implemented the architecture using Keras[3], a Python library that allows us to develop machine and deep learning models in an easy and fast way. The input layer contains a lookup matrix composed by the word dictionary index and its relative word embedding vector in order to convert the word indexes received from pre-processing. The word embedding vectors are extracted from a file with 300 dimensional vectors of 3 billion words[4].

The GRU layers of each network receives the sequence of word embeddings and encodes the input sentence into a 50-dimension vector. Recurrent weights are initialized using a random orthogonal matrix [12] and the internal weights of the GRU cells are initialized using *Xavier* algorithm [13] due to its capacity to define initial weights based on input and output units. This method of weights initialization follows the recommended parameters defined on the Keras library.

The weight structure W implemented for the recurrent neural networks is shared, shown in Figure 3, preserving the symmetry of the distance function [7]. The linear mapping G_w [6], shown in Figure 3, is applied over the inputs x_a and x_b, which relies on the shared weights W. Thus, both recurrent neural networks from the siamese architecture receive the same updates from the backpropagation algorithm.

C. Training Details

We apply Adadelta [14] algorithm for weights optimization during training since Adadelta can automatically decrease learning rate. Based on empirical tests, we set 0.5 to the initial learning rate for Adadelta. Furthermore, we employ gradient

[3]https://keras.io/

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

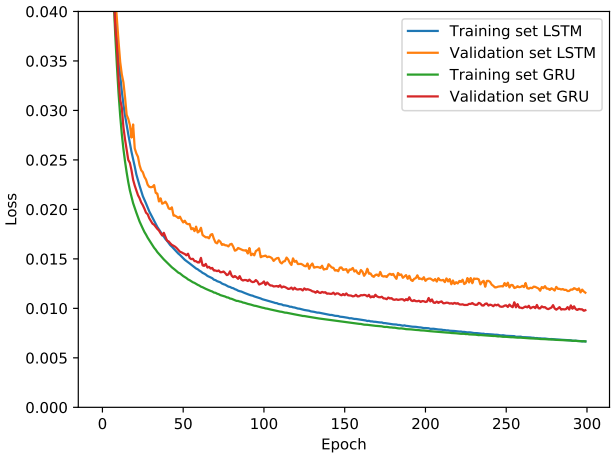

Fig. 4. Learning curves for training and validation sets, comparing performance in use of GRU and LSTM. This training execution uses parameter values defined in Section IV.

clipping strategy [4] with a threshold value of 1.5 in order to deal with vanishing and exploding gradients.

Since our siamese network have shared weights, both recurrent networks receive the same update from backpropagation, which is measured using the Mean Square Error loss function using only the predicted number and annotation of sentence pair. For each execution, network processes a mini-batch of 32 sentence pairs and full training is executed using 300 epochs. All these parameter values were defined based on empirical tests.

V. Experiments and Results

In this section, we describe the experiments and test scenarios executed over the SICK dataset, detailing the results using SemEval metrics. We use the SemEval metrics (Pearson/Spearman correlation and mean square error) to compare with other proposed methods for the same SemEval Task, which uses the same dataset.

A. Recurrent Layer Architectures

We compare two recurrent neural network architectures (LSTM and GRU) that address the vanishing/exploding gradient problem using gated structure on the SICK dataset. The motivation to make this comparison relies on Chung *et al* [11] work, which demonstrates that both architectures have equivalent performances in sequence modeling task, although GRU have less parameters than LSTM. Furthermore, this comparison is motivated by Mueller *et al* work [8], which surpassed state of the art using LSTM as a recurrent architecture to encode sentences in a siamese architecture. For such test case, we implemented one siamese network using GRU units and another one using LSTM units. Both implementations follow definitions described in Section IV.

In our tests, we noted that using the GRU architecture in the siamese network outperforms LSTM in all SemEval comparison metrics, as shown in Table I. Moreover, learning curves shown in Figure 4, evidence that GRU can deal better with unseen data due to a smaller loss in the execution over validation set than LSTM. These results motivate the use of GRU architecture on the final version of our work since it generalizes well using a small number of training sentence pairs.

TABLE I
Comparison of results using different recurrent architecture, using SemEval metrics to determine which architecture generalizes better on SICK test set.

Architecture	Pearson	Spearman	Mean Square Error
LSTM	0.7983	0.7492	0.3779
GRU	**0.8448**	**0.7902**	**0.3032**

TABLE II
Results of methods applied in semantic textual similarity task over SICK test set. Table is divided in three groups: first is top SemEval-2014 submissions, second is later works and third is the work that reaches state of the art.

Method	Pearson	Spearman	MSE
Illinois-LH (Lai *et al*w - 2014) [15]	0.7993	0.7538	0.3692
UNAL-NLP (Jimenez et al. - 2014) [16]	0.8070	0.7489	0.3550
Meaning Factory (Bjerva et al. - 2014) [17]	0.8268	0.7721	0.3224
ECNU (Zhao *et al* - 2014) [18]	0.8279	0.7689	0.3250
MaLSTM (Mueller *et al* - 2016)	0.8222	-	-
Siamese GRU Model (Ichida *et al* - 2017)	0.8448	0.7902	0.3032
Dependency Tree-LSTM (Tai*et al* - 2015) [19]	0.8686	0.8047	0.2606
MaLSTM +Syn Augmentation +Transfer Learning (Mueller *et al* - 2016)	**0.8822**	**0.8345**	**0.2286**

B. Comparative with SemEval Published Methods

We compare our work with the best works submitted to SemEval-2014 and others notable works. In Table II, we show that our approach outperforms all the best works submitted in SemEval-2014. Our work does not achieve state of the art when compared to the approaches from Mueller and Tai. Although our work is based on Mueller *et al* method, we do not use synonym augmentation techniques and transfer learning described in their work. However, Table II shows that our results surpasses MaLSTM method considering only the use of siamese neural network architecture without any extra techniques such as synonym augmentation.

We analyze individual results in determined sentence pairs, comparing a sentence to others contained in the SICK test set. Table III contains sentences and results of MaLSTM and Dependency Tree-LSTM extracted from Mueller *et al.* work [8], composed by Column S representing predicted values of our implementation of Siamese-GRU model, Column M representing predicted value of Mueller work using all of

TABLE III
COMPARISON AND RESULTS OF OUR SIAMESE GRU METHOD AGAINST APPROACHES OF MUELLER [8] AND TAI [19] USING SENTENCES CONTAINED IN SICK TEST SET. WE COMPARE SENTENCE IN BOLD WITH EACH SENTENCE CONTAINED IN TABLE GROUP.

Sentence	S	M	T
a woman is slicing potatoes			
- a woman is cutting potatoes	4.79	4.87	4.82
- potatoes are being slices by a woman	4.41	4.38	4.70
- tofu is being sliced by a woman	2.71	3.51	4.39
two men are playing guitar			
- the man is singing and playing the guitar	3.15	3.53	4.08
- the man is opening the guitar for donations and plays with the case	2.91	2.30	4.01
- two men are dancing and singing in front of a crowd	2.56	2.33	4.00

techniques described in his work and column T representing results from Tree-LSTM.

In this comparison, we noted that our model predicts values close to the results of state of the art [8], even though pearson correlation between our results and ground truth values is lower than Dependency Tree-LSTM method. Additionally, we noted that our approach can deal better at detecting the subject of sentences than Tree-LSTM approach, shown in the predicted similarity of the sentences "a woman is slicing potatoes" and "tofu is being sliced by a woman".

This comparison shows that our approach results in distance function that measures similarity values close to the best approaches in SICK test set. Although our results do not outperforms the state of the art method, comparing some individual predicted values to their work shows that our siamese neural network approach can efficiently results the semantic similarity, surpassing submitted methods of SemEval 2014 edition.

C. Verbal Voice Forms

In this test scenario, we explore the variation of verbal voice in a sentence, analyzing predicted similarity given a sentence in active voice with his respective passive voice form. Changing verbal voice does not imply in a change in sentence context, thus is expected that our approach results in high values in this scenario. We use sentence pairs that are not in SICK dataset due to verify how our method generalizes in unseen sentences during training execution. Moreover, we select different verb tenses and types in order to verify the results and how our approach react with these modifications.

Table IV shows the predicted semantic similarity of our siamese network implementation over sentences with different verb tenses and types. We note that our network are not sensitive in cases where main verb is altered due to voice form variation, resulting in a similarity below expected. For example, in sentence "Someone is painting the building wall." when we alter to his passive voice form, the verb "painting" changes to "painted". A simple way to resolve these limitations is generate enough training sentence pairs that varies the verbal voices of a individual sentence based on verbal tenses that resulted low values.

D. Paraphrase Sentences

Due to fact that a paraphrase is characterized by rewriting of a sentence using different words but maintaining the meaning, we empirically evaluate our siamese neural network using sentences with paraphrase relationship using Microsoft Research Paraphrase Corpus dataset [20]. This analysis shows that some sentences pairs classified as paraphrase resulted in a low similarity semantic value due to some sentences have extra pieces of information that the paired sentences does not. For example, in Table V, sentence "The DVD CCA then appealed to the state Supreme Court." does not inform if supreme court is US Supreme court, resulting a low predicted similarity.

This test scenario shows that our model have a limitation regarding context due to execute the training using individual sentences without information about preceding and following information. To deal with this limitation, we need to combine other reasonable alternatives such as Skip-Thought Vectors [21], which is a recurrent neural network that encodes a sentence into a vector trained reconstructing the immediately preceding and following sentences.

VI. RELATED WORK

Mueller and Thyagarajan [8] proposes predict the semantic similarity through learning a metric, which relies on a siamese neural network architecture. First, using word embeddings to represent words contained in input sentences, their work use a recurrent neural network to learn a sentence encoding through a distance function between two inputs. Second, their work uses a synonym augmentation technique to expand the SICK dataset, generating 10,022 additional training examples replacing random words of original sentence with one of their synonyms extracted from Wordnet [22]. Although is similar to Mueller and Thyagarajan's approach, our works is differs by the fact of use of GRU as gating recurrent architecture, which we show that can generalize better in SICK dataset than LSTM in Section V.

Tai, Socher and Manning [19] propose a generalization of recurrent neural network, processing sequences of words in a tree-structured LSTM. Their work focuses on generating an encoder to represent information of a sentence more efficiently than a sequentially network default. Their work relies on a dependency parser [23] to represent the input sentences hierarchically. Our approach is simpler, using a sequentially recurrent network that uses word vectors as input representation, which is extracted from a pre-trained word embedding model.

Aires and Meneguzzi [24] propose an algorithm to measure the semantic similarity between sentences using the Wu-Palmer Distance of words. They use such measure to detect semantic similar norm actions within contract texts, mensuring similarity using a word-level distance measure. Although their work obtains satisfactory results, our method explores a more complex approach to measure semantic similarity in diverse contexts, considering more items contained in a sentence-level than a word-level.

TABLE IV
RESULTS WITH SEMANTIC SIMILARITY PREDICTED IN SENTENCE PAIRS ASSOCIATED WITH THE RESPECTIVE VERBAL TYPE/TENSE. TABLE IS ORDERED BY PREDICTED SIMILARITY.

Active Voice Form	**Passive Voice Form**	**Verb Tense/Type**	**Similarity Predicted**
Michael Jordan bought the Bobcats team.	The Bobcats team was bought by Michael Jordan.	Simple Past	0.96
Alex writes a small book.	A small book is written by Alex.	Simple Present	0.94
John Doe had bought a Ford Fiesta.	A Ford Fiesta had been bought by John Doe.	Past Perfect	0.94
John could have bought this house.	This house could have been bought by John.	Modal	0.94
Bill would have won the fight.	The fight would have being won by Bill.	Modal	0.89
She can create a python program.	A python program can be created by her.	Modal	0.79
A woman is writing a letter.	A letter is being writen by a woman.	Present Continuous	0.77
Someone is painting the building wall.	The building wall is being painted by someone.	Present Continuous	0.76
Rita was writing a letter.	A letter was being written by Rita.	Past Continuous	0.73

TABLE V
TABLE WITH PAIR OF SENTENCES CONTAINED IN MICROSOFT RESEARCH PARAPHRASE CORPUS CLASSIFIED AS PARAPHRASE RELATIONSHIP.

Sentence A	**Sentence B**	**Similarity Predicted**
Revenue inthe first quarter of the year dropped 15 percent from the sameperiod a year earlier.	With the scandal hanging over Stewart's company, revenue the first quarter of the year dropped 15 percent from the same period a year earlier.	0.83
The DVD CCA then appealed to the state Supreme Court.	The DVD CCA appealed that decision to the U.S. Supreme Court.	0.69
But he added group performance would improve in the second half of theyear and beyond.	De Sole said in the results statement that group performance would increase in the second half of the year and beyond.	0.77

VII. CONCLUSION AND FUTURE WORK

We have implemented a Siamese neural network architecture to measure semantic similarity between two sentences and evaluated it through different test scenarios, comparing with results of related works. Our work achieves results close to the state of the art leveraging a simplified neural network architecture, which generalizes beyond few sentence pairs. Word embedding help us to obtain an efficient input representation, retrieving semantic and syntactic information in a word level. Thus, our work does not require an extensive manual feature generation due to use a existing pre-trained model, dispensing the linguistic information of the sentences.

In this work, we realize test scenarios that reveals limitations of our approach, which motivates following future works. First, we intend to explore the semantic representation generated by the recurrent neural network of siamese architecture, which can be applied a dimension reduction to create a informative visualization of learned encoding. Second, we aim to enhance our work for considering the preceding and following sentences following the Kiros *et al* work [21], capturing more information about context of the information.

Finally, we intend to apply dataset augmentation techniques, which deals with size-limitations of labeled datasets in Mueller work [8]. This task contains a difficulty that lies in deciding which words can be replaced and what synonym options are a valid replacement, in order to maintain context of original sentence.

ACKNOWLEDGMENTS

Felipe thanks CNPq for partial financial support under its PQ fellowship, grant number 305969/2016-1.

REFERENCES

[1] M. Marelli, L. Bentivogli, M. Baroni, R. Bernardi, S. Menini, and R. Zamparelli, "Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment," in *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, 2014, pp. 1–8.

[2] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[3] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[4] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[5] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[6] B. Kulis *et al.*, "Metric learning: A survey," *Foundations and Trends® in Machine Learning*, vol. 5, no. 4, pp. 287–364, 2013.

[7] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 539–546.

[8] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity." in *AAAI*, 2016, pp. 2786–2792.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[12] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," *arXiv preprint arXiv:1312.6120*, 2013.

[13] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. [Online]. Available: http://proceedings.mlr.press/v9/glorot10a.html

[14] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.
[15] A. Lai and J. Hockenmaier, "Illinois-lh: A denotational and distributional approach to semantics," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 329–334.
[16] S. Jimenez, G. Duenas, J. Baquero, and A. Gelbukh, "Unal-nlp: Combining soft cardinality features for semantic textual similarity, relatedness and entailment," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 732–742.
[17] J. Bjerva, J. Bos, R. Van der Goot, and M. Nissim, "The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 642–646.
[18] J. Zhao, T. Zhu, and M. Lan, "Ecnu: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 271–277.
[19] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *arXiv preprint arXiv:1503.00075*, 2015.
[20] B. Dolan, C. Brockett, and C. Quirk, "Microsoft research paraphrase corpus," *Retrieved March*, vol. 29, p. 2008, 2005.
[21] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Advances in neural information processing systems*, 2015, pp. 3294–3302.
[22] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
[23] D. Chen and C. Manning, "A fast and accurate dependency parser using neural networks," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 740–750.
[24] J. P. Aires, D. Pinheiro, V. S. d. Lima, and F. Meneguzzi, "Norm conflict identification in contracts," *Artificial Intelligence and Law*, vol. 25, no. 4, pp. 397–428, Dec 2017.