Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 145 (2018) 464–470

Procedia Computer Science

www.elsevier.com/locate/procedia

Postproceedings of the 9th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2018 (Ninth Annual Meeting of the BICA Society)

A Neural Network model for the Evaluation of Text Complexity in Italian Language: a Representation Point of View

Giosué Lo Bosco[a], Giovanni Pilato[b], Daniele Schicchi[a,*]

[a] *Dipartimento di Matematica e Informatica, Univerisitá degli Studi di Palermo, Palermo , Italy*
[b] *ICAR-CNR - National Research Council of Italy, Palermo, Italy*

Abstract

The goal of a text simplification system (TS) is to create a new text suited to the characteristics of a reader, with the final goal of making it more understandable.The building of an Automatic Text Simplification System (ATS) cannot be separated from a correct evaluation of the text complexity. In fact the ATS must be capable of understanding if a text should be simplified for the target reader or not. In a previous work we have presented a model capable of classifying Italian sentences based on their complexity level. Our model is a Long Short Term Memory (LSTM) Neural Network capable of learning the features of *easy-to-read* and *complex-to-read* sentences autonomously from a annotated corpus created specifically for text simplification. In this paper we further investigate on the role of the text representation, i.e. how different ways of representing the input text can affect the accuracy of the proposed system. In detail, we will use our Neural Network model for evaluating the sentence complexity using different kind of representations such as GloVe, Word2vec, FastTex and a new one based on a representation learning scheme.

Keywords: Text Simplification; Natural Language Processing; Deep Neural Networks; Evaluation Sentence Complexity; Sentence Classification.

[*] Corresponding author.
E-mail address: daniele.schicchi@unipa.it

10.1016/j.procs.2018.11.108

1. Introduction

Text simplification (TS) is a process that aims at reducing the linguistic complexity of a text by modifying its syntactic structure and substituting lemmas. The result of TS is a new text that keeps the original meaning, but that is more easily readable and understandable.

It has been shown the utility of TS for different classes of people, for example, who have language disabilities, who are not mother tongue or who have a low educational level. Children affected by deafness needs help for facing reading difficulties [12, 15] or people affected by dyslexia have to face comprehension difficulties in reading infrequent and long words. Another aspect that has to take into account is the high percentage of people with low literacy skills that are unable to understand common texts. For example, Italy is one of the countries with a considerable number of people with low linguistic competencies [14].

The building of an Automatic Text Simplification System (ATS) cannot be separated from a correct evaluation of the text complexity; in fact, an ATS system must be capable of understanding if the text should be simplified for the target reader or not.

Although many researchers have tackled the automatic evaluation of sentence complexity for the English language, there is a lack for the Italian one. Since the diversities between the two languages, not all the results obtained for the English language can be reproduced for the Italian one thus, it is necessary to further research for what concern TS for the specific language.

In our previous work [10] we have presented a model capable of measuring the complexity level of an Italian sentence. Our model is a Neural Network (NN) based on Long Short-Term Memory (LSTM) [8] units capable of learning the features of *complex-to-read* and *easy-to-read* sentences autonomously from an annotated corpus [3].

In this paper, we further investigate the role of the text representation, i.e., how different ways of representing the input text can affect the accuracy of the proposed system. In detail, we will use our Neural Network model for evaluating the sentence complexity using different kind of representations such as GloVe [16], word2vec [13], fastText [2] and a new one based on a representation learning scheme.

The paper is structured as follow: in section 2 we describe the state of art of text evaluation for Italian language, in section 3 we introduce the model we have used and the representation schema, in section 4 we show the methodology used for testing the model and results, in section 5 and in section 6 we give our consideration about the results and conclusions.

2. Related Work

Historically, measures for the text complexity have relied on a set of structural text features like the length of the text, the number of syllable per words, the number of characters of the text and so on. The most common measures to score the complexity of an Italian text are Flesch-Vacca [5] and GulpEase [11]. The former is derived from the Flesch-Kincaid measure [9] that use as a parameter of evaluation the average number of syllables per word and the average number of words per sentence, while the second one is based on the average number of characters per words, the average number of words per sentence and the number of sentences. It has been shown that these classes of indexes are inadequate to cover all aspects of the language complexity, in fact, for example, both indexes evaluate more difficult to read a longer text, but it could be not true. A longer text could contain further pieces of information that may help the reader for understanding the entire semantic content.

Furthermore, other class of people, such as those with cognitive impairments, need a system capable of considering another level of text complexity that is not only related to words, but that takes into account the syntactic structure of the sentence also of his lemmas [17].

The most common complex system capable of measuring the complexity of a sentence written in the Italian language is READ-IT [4]. READ-IT is a classifier based on Support Vector Machine (SVM) which considers many linguistic features to understand what is the degree of complexity of a sentence. The SVM model has been trained using two corpora ”La Repubblica”[1] which contains many sentences considered *difficult-to-read* [4] and ”Due Parole”[2] which contains only *easy-to-read* [4] sentences written for a low literacy people by a pool of linguistic expert. The SVM receives as input a vector that is the combination of linguistic measures calculated on the text which are related to the sentence lexical aspects (e.g. the presence of *easy* terms in the sentence), the sentence morphology (e.g lexical density and verbal mood) and the syntactical aspect of the sentence (e.g the depth of parse tree and distribution of subordinate clause).

3. Proposed Methodology

Our model [10] is capable of learning autonomously the features of *easy-to-read* and *difficult-to-read* sentences analyzing an annotated corpus [3]. It is able to classify sentences in two classes: *complex-to-read* and *easy-to-read* giving a *score* for each sentence that represent the confidence level of the model during the decision process.

The system is characterized by the adoption of a particular class of NNs called Recurrent Neural Networks (RNNs) [6] which fits well the problem of analyzing data sequences. RNN belongs to the class of Neural Networks [1] which, in the recent past have shown good results in many different linguistic fields.

It is well known that a sentence can be interpreted as a sequence of words, that represent the lexical aspects, and punctuation symbols that represent the syntactical aspects of the original sentence. This kind of networks can examine the symbols of the sequence step by step but taking into account what it has been previously analyzed. Thus, in our case, the network can make a decision that is a function of lexical and syntactical aspects of the sentence.

3.1. Preprocessing

During the preprocessing phase, each sentence is divided into a sequence of tokens where a token is either a word or a punctuation symbol. This kind of splitting process is well known in the literature, and it guarantees representation of the sentence without loss of information, even if stop-words and punctuation are often neglected, in our case all kind of symbols could affect the sentence complexity.

After the division of the sentence in a sequence of tokens, it is necessary to represent each token as a vector V in an n-dimensional space that can be evaluated by the network. We have decided to transform each token in such a vector V in a different way investigating if the representation of the tokens can heavily affect the result of our model trying to respond to the question if the evaluation of a sentence complexity in the Italian language is mainly a representation problem.

We have chosen for the vector representations of the tokens GloVe [16], Word2vec [13], FastText [2] and a new one based on a representation learning scheme. All these representation methodologies map each token in a 300-dimensional space vector; thus at the end of the preprocessing, a sentence is a sequence of a 300-dimensional vector that represents the meaning and the structure of the sentence.

3.1.1. Word2vec

Word2Vec is the first representation model based on a Neural Network. These are two-layer neural networks that are trained to reconstruct linguistic contexts of words.

The idea is that the meaning of a word can be understood by other words that are in the same context. If two words have a similar context, then they probably are related, or they have a similar meaning. Using this idea, word2vec is used to find related and unrelated concepts or compute the similarity between two words.

For our experiment we have used a pretrained word2vec model[3] in which each word and punctuation symbols are mapped to a vector in a 300-dimensional space.

[1] www.repubblica.it

[2] www.dueparole.it

[3] github.com/Kyubyong/wordvectors

3.1.2. GloVe

The vector representation of a word or punctuation symbol is obtained by unsupervised learning algorithm trained aggregating global word-word co-occurrence statistics from a corpus and the resulting representations show linear relations between substructures into the word vector space.
GloVe [16] is a log-bilinear model with a weighted least-squares objective. The idea underlying the model is that the ratios of word-word co-occurrence probabilities could encode a form of meaning. The training objective of GloVe is to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence. The resulting word vectors perform very well on word analogy tasks.
In our experiment, we have used a pretrained gloVe model where, as in word2vec model, each word and punctuation symbol is mapped to a vector in a 300-dimensional space.

3.1.3. FastText

FastText is a library for efficient learning of word representations and sentence classification. The FastText model takes into account the morphology of the word and his internal structure. It considers as features not only the word itself but also the bag of characters that compose it. We have used FastText pretrained on Common Crawl [18] and Wikipedia that is a map from a word or punctuation symbol to a space of 300-dimensional vectors.

3.1.4. Auto-Learning

Although the low number of pairs of sentences in our corpus, we have tried to insert an embedding layer able to build his own representation of tokens in a 300-dimensional space. In particular, we encode each word and punctuation symbol using a well known one-hot encoding with a dictionary of size 50.000. In this manner, each sentence is evaluated as a sequence of numbers in which each number is the index of the original word inside the dictionary. The network examines the sequence of numbers and calculates vectors into 300-dimensional space trying to minimize the loss function.

3.2. Architecture

The Network architecture is based on LSTM [8] artificial neurons. LSTMs have been successfully used to face many Natural Language Processing problems. A good peculiarity of LSTM units is its abilities to face the problem of vanishing gradient [6] and of remembering the dependencies among elements inside a sequence which are distant from each other.
The first layer of the network that analyzes the representations of words and punctuation symbols are built on 512 LSTM units. The outcome of this layer is then processed by a fully connected layer composed of two neurons adopting the softmax as activation function. Finally, we have applied L_2 regularization. The network architecture is shown in figure 1.
The softmax function represents the degree of confidence with which the model assigns a sentence to one of the two classes, and the score could be interpreted as a cumulative score that measures the complexity of the sentence after the analysis of lemmas and syntactic structure.

3.3. Parameters

The research of parameters has been done through a series of experiments. We have observed that the network reached the goal of our study limiting the source sentences to 20 tokens and training the network for 8 epochs. The loss function used is the well-known *cross-entropy* and the optimization algorithm we have chosen is the RMSPROP [7] algorithm on balanced minibatch of size 25. For what concerns the choice on the number of tokens, we have not observed valuable improvements choosing a number greater than 20.

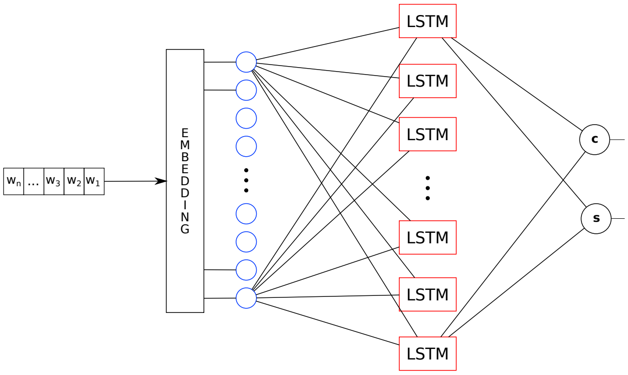

Fig. 1. The Neural Network model capable of classifying sentences in *easy-to-read*(s output) and *complex-to-read*(c output). The input is examined by an embedding layer which transforms each token into a 300-dimensional vector. The LSTM layer analyze the sequence of vectors and the final dense layer with softmax activation function chose the class of the input sentence.

4. Experiments and Results

4.1. Corpus

The ideal corpus would be a set of sentences labeled with their corresponding level of difficulty that is not available for the Italian language. The lack of resources complicates the problem of TS for the Italian language thus a hard task is to find a corpus suited to the methodology that a researcher intends to use. At the time of writing of this work, there is only one corpus [3] suited for training Neural Network algorithms that we have used to conduct our research for the evaluation of the sentence complexity.

The corpus contains about 63.000 pairs of annotated sentences in which each original sentence has the corresponding simplification in it. The paired sentences represent structural transformations that identify how to simplify a sentence, thus all the simplified sentences can be considered *easy-to-read* and can be used for training the sentence classification algorithm. Inside the corpus there are many simplification rules such us *deletion* of some words that make the sentence more difficult to understand, the *insertion* of more informative content that can help the comprehension of the sentence or the *changing* of the verbal mood and verbal tense that could increase the complexity of the sentence.

The analysis of the corpus allows the Network to infer lexical and syntactic peculiarities that characterize *easy-to-read* or *difficult-to-read* sentences.

4.2. Experiments

We have conducted a series of tests using the K-FOLD cross-validation (K-FOLD) with $K = 10$ to evaluate the contribution of different representation scheme to classify sentences into the two classes: *difficult-to-read* and *easy-to-read*. K-FOLD cross-validation is a validation method useful when there are not enough data for trying and test the system.

In our case, we have only 63.000 pairs of sentences which means that for obtaining good results we must use most of the dataset for training the network. Thus the use of most of the dataset leaves not enough data for experimenting with our model. K-FOLD allows of overcoming the problem assessing the abilities of our model testing it on an independent dataset. The functioning of a K-FOLD method is to partition randomly the dataset into K subsets of equal size: the method select K-1 subsets that are used to train the model while the last one is used to validate it. We have trained K models considering the two classes of sentences.

To quantify the obtained results we have calculated, for each iteration of K-FOLD and each representation schema, the standard measures for classification algorithms: Precision, Recall, True Positive Ratio (TPR) and True Negative Ratio (TNR). Recall and Precision represent respectively the portion of the positive elements class that the model has correctly classified and how many elements the model has classified as positive while they belong to the negative class. TPR[4] and TNR express information about how effective is the classifier for identifying the correct class for elements of both classes. The final result is the average of each measure on K iterations performed. Table 1 shows the results obtained by our Network.

Table 1. The average of Precision, Recall, True Positive Ratio and True Negative Ratio on 10 iteration performed.

Representation Method	Recall	Precision	True Positive Ratio	True Negative Ratio
Auto-learning	**0.844**	**0.869**	**0.844**	**0.873**
Word2Vec	0.841	0.858	0.841	0.860
GloVe	0.843	0.862	0.843	0.864
FastText	0.838	0.868	0.838	0.871

5. Discussion

In this paper, we are trying to investigate if the problem of classifying sentence based on their complexity can be affected by different type of word and punctuation symbols representation.

In detail, our previous work has shown good results using a neuronal system to address this classification problem, capable of inferring the rules that identify the peculiarities of *easy-to-read* and *complex-to-read* sentences analyzing only the components of the sentence without taking into account other features such as the sentence parse-tree, the presence or absence of easy words and so on.

Although the representation schema used calculate the embedding space in different manners and using different corpora, the result we have obtained are very similar. This suggests that the problem of evaluating the complexity of a sentence is representation independent.

Furthermore, the *auto-learning* representation schema slight overcome all other competitors schema. The problem with the auto-learning is that since there are few pairs of sentences inside the corpus the vocabulary that the model has learned is not very rich and it is probably adequate to this specific corpus.

Nonetheless, the powerful of the *auto-learning* representation is shown for the sentence evaluation problem for Italian language and the same method can be used for bigger corpora.

Our work can be considered as a preliminary investigation to understand if it is more convenient making effort for finding a better representation schema or create a better model. In our future work we will carry out experiments on other ways to represent the words and punctuation symbols and we will hopefully extend test on other corpus in order generalize the obtained result. At this moment, experiments suggest that for creating an ATS would be a good idea spend more time in trying different models rather than for choosing the best representation schema.

6. Conclusion

In this paper, we have looked into the problem of evaluating the complexity of an Italian sentence deepening how a representation schema can affect the results of the model.

We have used the same neuronal model, which works well to address the classification task, and we have tried different representation schema. The results of our tests show that the measures that identify the correctness of the model are very similar among all the representation methods suggesting that the problem of evaluating the text complexity is more related to the nature of the model than to the representation method of the sentence tokens.

[4] TPR is calculated in the same way of RECALL.

References

[1] Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Oxford University Press, Inc., New York, NY, USA.

[2] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., 2016. Enriching word vectors with subword information. CoRR abs/1607.04606.

[3] Brunato, D., Cimino, A., Dell'Orletta, F., Venturi, G., 2016. Paccss-it: A parallel corpus of complex-simple sentences for automatic text simplification, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics. pp. 351–361. URL: http://www.aclweb.org/anthology/D16-1034, doi:10.18653/v1/D16-1034.

[4] Dell'Orletta, F., Montemagni, S., Venturi, G., 2011. Read-it: Assessing readability of italian texts with a view to text simplification, in: Proceedings of the second workshop on speech and language processing for assistive technologies, Association for Computational Linguistics. pp. 73–83.

[5] Franchina, V., Vacca, R., 1986. Adaptation of flesh readability index on a bilingual text written by the same author both in italian and english languages. Linguaggi 3, 47–49.

[6] Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

[7] Hinton, G., Srivastava, N., Swersky, K., 2012. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent.

[8] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9, 1735–1780.

[9] Kincaid, J.P., Fishburne Jr, R.P., Rogers, R.L., Chissom, B.S., 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical Report. Naval Technical Training Command Millington TN Research Branch.

[10] Lo Bosco, G., Pilato, G., Schicchi, D., 2018. A recurrent deep neural network model to measure sentence complexity for the italian language, in: in press: International Workshop on Artificial Intelligence and Cognition, 6th Edition, Palermo, Italy.

[11] Lucisano, P., Piemontese, M.E., 1988. Gulpease: una formula per la predizione della difficoltà dei testi in lingua italiana. Scuola e città 3, 110–124.

[12] Marschark, M., Spencer, P.E., 2010. The Oxford handbook of deaf studies, language, and education. volume 2. Oxford University Press.

[13] Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. CoRR abs/1301.3781.

[14] OECD, 2013. Inchiesta sulle competenze degli adulti primi risultati.

[15] Paul, P.V., 2009. Language and deafness. Jones & Bartlett Learning.

[16] Pennington, J., Socher, R., Manning, C.D., 2014. Glove: Global vectors for word representation, in: Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543.

[17] Siddharthan, A., 2014. A survey of research on text simplification. ITL-International Journal of Applied Linguistics 165, 259–298.

[18] www.commoncrawl.org, .