A Recurrent Deep Neural Network Model to measure Sentence Complexity for the Italian Language

Giosué Lo Bosco[1][0000−0002−1602−0693], Giovanni Pilato[2][0000−0002−6254−2249], and Daniele Schicchi[1][0000−0003−0154−2736]

[1] Dipartimento di Matematica e Informatica, Universitá degli studi di Palermo, ITALY
{giosue.lobosco,daniele.schicchi}@unipa.it
[2] ICAR-CNR - National Research Council of Italy, Palermo, ITALY
giovanni.pilato@icar.cnr.it

Abstract. Text simplification (TS) is a natural language processing task devoted to the modification of a text in such a way that the grammar and structure of the phrases is greatly simplified, preserving the underlying meaning and information contents. In this paper we give a contribution to the TS field presenting a deep neural network model able to detect the complexity of italian sentences. In particular, the system gives a score to an input text that identifies the confidence level during the decision making process and that could be interpreted as a measure of the sentence complexity. Experiments have been carried out on one public corpus of Italian texts created specifically for the task of TS. We have also provided a comparison of our model with a state of the art method used for the same purpose.

Keywords: Text Simplification · Natural Language Processing · Deep Neural Networks.

1 Introduction

Text simplification (TS) is a process that aims at reducing the linguistic complexity by modifying syntax structure and substituting lemmas in a text. The result of TS is a new text that keeps the original meaning but that is more easily readable and understandable.
TS is useful for many different kinds of people such as who are not mother tongue, or have language disabilities, with low educational level and so on. For example, children affected by deafness have to face many reading difficulties caused by linguistic problems arisen in their youth [17, 19] or people affected by dyslexia have comprehension difficulties in reading infrequent and long words. Furthermore, although the increasing investments for school and instruction are helping the growing of personal culture, there still is a huge percentage of people with low literal 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 [18].
Although, there has been a lot of research on TS for English language, there is a lack of resources for the Italian language that could be used to build TS systems. Algorithms that work well for English language have lead to poor performance for the Italian one underling the differences between the two languages. Despite these difficulties many works have been made trying to face different NLP problems [1, 6, 7] suggesting that much remains to be done for what concern the automatic analysis of Italian texts.
In this paper, we give a contribution to the TS field using neural networks for developing a system capable of classifying Italian sentences in 2 classes according to their complexity. The system gives a score that represents the level of confidence during the decision process and that could be interpreted as a measure of a sentence complexity for low literacy skills readers.
In the domain of TS, as underlined in the work by Shardlow [20], words like *complex* and *simple* should be used keeping in mind that their meaning is relative to each other and the difficulty of a sentence is related to a determined class of people that could have different needs. Unfortunately, since the nature of the corpus we have used, our system is not specialized for any class of people but acquire a more general knowledge about *complex* and *simple* meaning.
The paper is organized as follow: in section 2 we describe some of the works related to TS, in section 3 we will describe the system and our approach of facing the problem, in section 4 we will explain the methodology of carrying out the tests and results, in section 5 we will give conclusion.

2 Related Works

The problem of evaluating a sentence complexity is a research field that has been faced using different methodologies. Historically, measures have relied on a set of structural text features like the length of the sentence, the number of words syllable or the number of characters. For example, for the case of Italian language Flesch-Vacca [9] and GulpEase [16] are the most common used tools to score the complexity of a text. The former is an adaptation of Flesch-Kincaid measure [15] that is function of the average number of syllables per word and the average number of sentence words, while the latter is based on the average number of word characters and the average number of words per sentence. Unfortunately, it's a common opinion that these classes of indexes are not able to cover the multiple aspects of complexity, in fact, for example, both indexes consider longer sentences as more difficult to read, which could be not the real case. The weak efficiency of such kind of measures has led to the development of new measures that take into account a *simple* words dictionary [5].
Apart from what has been produced so far in the contest of TS, it needs to consider another level of text complexity that is not only words related, but take into account the syntactic structure of the sentence in addition of his lemmas. In fact, some syntactical structures causes a loss of meaning for person with cognitive impairments such as aphasia. Thus, a specific assistance

for these people is the reorganization of the sentence structure which makes it easier to comprehend. A Neural Network (NN) model based on Long Short Term Memory (LSTM) [13] units could be able to evaluate both syntactical and lexical aspects of complexity, learning the features of *complex* and *simple* sentences autonomously from data.
Nowadays, the most important index for assessing sentence complexity is READ-IT [8]. READ-IT is a classifier based on Support Vector Machine (SVM) which considers many linguistic features that represent the complexity of the sentence. READ-IT has been training using "La Repubblica" as examples of complex sentences and "Due Parole" as example of simple sentences. The SVM receives as input a vector that represents linguistic features which are divided in three classes: *Lexical Features*, *Morpho-syntactic Features* and *Syntactic Features*. Each one of these classes include many measures which describe the sentence under different points of view. For example, *Lexical* class includes elements related to the sentence lexical aspects such as the ratio between the number of lexical types and the number of tokens or the presence of *easy* terms in the sentence. *Morpho-syntactic* class contains features that look into the morphology and syntax of the sentence measuring, amongst other, lexical density and verbal mood. Finally, *Syntactic* class contains features that take into account elements only related to the sentence's syntax such as the depth of parse tree, distribution of subordinate clause, distribution of main clause and so on.

3 Proposed methodology

The proposed methodology is able to understand the rules that characterize the difficulty of a sentence. It belongs to the class of Neural Networks [2] which, in the recent past, have shown good results in many different linguistic fields.
Our system is based on a particular class of NNs called Recurrent Neural Networks (RNNs) [10] which fits well the problem of analyzing data sequences. This kind of networks are able to examine the symbols of the sequence step by step but taking into account what it has been previously analyzed. Thus, the output of the network is function of sequence elements but also of their structure.
From the Natural Language Processing (NLP) point of view the sentences can be structured as a sequence of words and punctuation in which their positions identify the syntactic structure. In this context, a RNN is able to take into account both lemmas and syntactic structure to establish the complexity of the sentence.

3.1 Preprocessing

A sentence is divided into a sequence of tokens. In our model, tokens are words and punctuation symbols. Splitting a sentence as sequence of tokens is a well known technique for the representation of a sentence. Even if stop-words and

punctuation are often neglected, in our case all kind of words could affect the sentence complexity.
Because of the low number of pairs of sentences in our corpus, we have avoided to insert an embedding layer able to build his own representation of tokens in a n-dimensional space, choosing to use a pre-trained word representation. In particular, each token is converted in 300-dimensional vector through the use of the dictionary [11]. The authors have used FastText [3], a library for efficient learning of word representations and sentence classification, trained on Common Crawl [21] and Wikipedia to create a pre-trained word vector representation for 157 languages, including Italian.
At the end of preprocessing, the sentence is a sequence of 300-dimensional vector that represents the meaning and the structure of the sentence.

3.2 Architecture

Our Network architecture is based on LSTM [13] which has been widely used for many sequence modeling tasks, thanks to its abilities of facing the problem of vanishing gradient [10] and of remembering the dependencies among elements inside a sequence which are distant from each other.
Each element of the sequence, in our case the representations of words and punctuations, is analyzed by a layer of 512 LSTM units. The outcome of this layer is then processed by fully connected layer composed by 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 expresses the probability that a sentence belongs to one of the two classes and it might be interpreted as a cumulative score that measures the complexity of the sentence taking into account both lemmas and syntactic structure.

3.3 Parameters

We have observed good results when limiting the source sentences to 20 tokens and training the network for 8 epochs.
The loss function used is the well known *cross-entropy*, that has been minimized choosing the RMSPROP [12] algorithm on balanced minibatch of size 25.
For what concerns the regularization L_2 we have used a weight value of 0.01.
The parameters of the network have been obtained through a set of trials. In detail, we noted that the best results are obtained when training the network for 8 epochs, that if exceeded cause overfitting. For what concerns the choice on the number of tokens, we have not observed valuable improvements choosing a number greater than 20.

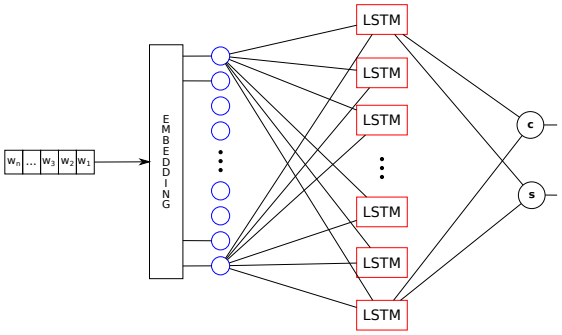

Fig. 1. The RNN model based on 512 LSTM neurons. A sentence s is evaluated as a sequence of words and punctuation w_i with $i = 1, 2, \ldots, n$. Each w_i is represented in a vector form by using of an external dictionary called *embedding*. **c** and **s** represent the classes of our sentences, they are the complementary outputs of softmax function.

4 Experiments and Results

4.1 Corpus

Since there is a lack of corpus for the Italian language suited for training machine learning algorithms we have used, to our knowledge, the biggest available dataset created for italian text simplification [4].
The corpus contains about 63.000 pairs of sentences in which each original sentence has the corresponding translation in its simplified form. The paired sentences containing structural transformations that identify how to simplify a sentence. Although the corpus has been created using an automatic procedure, it has been tested deeply with the help of the humans specialists. Some of simplification rules inside the corpus are *deletion* or *substitution* of words from a source sentence that make it difficult to understand, *insertion* of other words that can explain better the meaning of the sentence, *reordering* of *complex* sentence words in which lemmas position is changed in order to make the sentence more easily understandable.
We have trained the NN model used as *simple* sentences examples all the simplified sentences in the corpus, while, the others are used as *complex* sentences examples. The set of sentences allow the NN to understand what are the common patterns in the *complex* class and in the *simple* class.

4.2 Experiments

Since few pairs of sentences are present in our corpus, we decided to use the K-FOLD cross-validation (K-FOLD) with $K = 10$ to evaluate our approach. K-FOLD is a validation method for assessing the abilities of a statistical model in order to generalize his knowledge to an independent dataset. It partitions randomly the dataset into K equal sized subsets: 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 two classes of sentences: *in need of simplification (class positive)*, *simplified (class negative)*.
To quantify the obtained results we have calculated Precision, Recall, True Positive Ratio (TPR) and True Negative Ratio (TNR) for each iteration of K-FOLD. Recall and Precision give information about the percentage of positive class elements that the model is able to correctly classify and how many times it makes mistakes labelling an element as belonging to the positive class. TPR[3] and TNR express informations about how effective is the classifier for identifying the correct classes for elements of both classes. Finally, the results have been averaged on the K executed iterations. Table 1 shows the results obtained by our Network.

RECALL	PRECISION	True Positive Ratio	True Negative Ratio
0.83	0.86	0.83	0.87

Table 1. Measures of Recall, Precision, True Positive Rate, True Negative Rate.

To deeper understand the potentiality of our system, we have compared its performance with READ-IT [8]. Both our network and READ-IT are created to classify sentences based on their features. In detail they offer a score that is the sentence probability of belonging to one of the two classes. A score having a value close to 1 represents a high probability that the sentence needs to be simplified, otherwise a score close to 0 identifying no need of simplification.
Note that the READ-IT tool outputs only the sentence probability of beeing *complex*. For this reason, it is possible to compute an accuracy measure such as precision, recall, TPR and TNR only when setting a threshold value for the probability. Due to the difficulty in estimating such value, we have decided to compare our method with READ-IT by using the Area Under the Receiver Operating Characteristic (AUROC) curve [14]. The receiver Operating Characteristic (ROC) curve is a graphical plot that shows the classification abilities of a binary classifier when its discrimination threshold changes. If T is the set of thresholds that contains equal separated elements from 0 to 1, $\forall t \in T$ the ROC curve plots True Positive Rate (on the y axes) and False Positive Rate (in the x axes) calculated taking t as discrimination threshold. The AUROC represents the expectation that a uniformly drawn random positive is ranked before a uniformly drawn random negative.

[3] TPR is calculated at the same way of RECALL.

4.3 Discussion

The comparison in Figure 2 shows high values of AUROC for both models; in details our model achieves performances comparable to the READIT ones by showing noteworthy results for the same corpus with the same conditions.

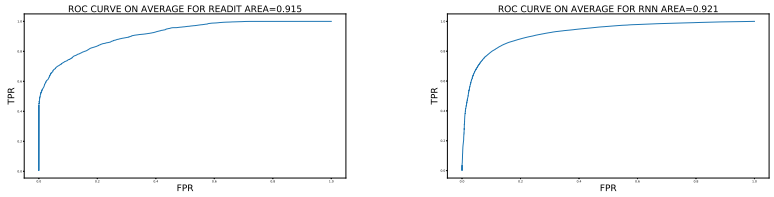

Fig. 2. On left the average ROC curve obtained by the mean of ROC curves out K iterations for READIT with AUROC=0.915. On right the average ROC curve obtained by the mean of ROC curves out K iterations for RNN with AUROC=0.921.

Although our system and READ-IT reach almost the same results they are deeply different. The main difference is due to the input data that the models accept. Our model is able to differentiate the classes of sentences taking as input the only raw texts. Otherwise, READ-IT needs to calculate many measures that are given as input to the SVM model.
Our idea is to build an easy-to-use model able to learn, on its own, linguistic features that identifies *simple* and *complex* objects building its knowledge on the dataset of annotated italian sentences. In addition, we think that the use of NN as the main model for understanding the complexity of sentences can better emulate the reasoning of human beings.
This model is open to a variety of application but we think that one of the most important is as a measure for reliable evaluation of automated text-simplification methodologies.
The lack of our model is related, like all NN-based approaches, to the difficulty of understanding the behavior of the NN. Its structure does not allow to understand which are the features of a sentence that make it *complex*, hence at this moment the model is not able to advice actions for simplify the sentence.

5 Conclusion

We have introduced a neuronal system able to automatically detect features that make italian sentences *complex* or *simple* for low literacy skills readers. The approach is completely data driven and makes use of raw text only. The main application of the system is that it provides a measure to quantify the complexity of a sentence in order to detect if a simplification is needed. In

addition, our system can be used as an embedded part of another system as a stand-alone module.
According to our test the system represents a good alternative to measure the sentence complexity level since its performance are coherent with the READ-IT ones. In spite of the system is not able to advice a simplification strategy we believe that it could represents a core component of a more complex system able to automatically simplify a generic text.

References

1. Alfano, M., Lenzitti, B., Lo Bosco, G., Perticone, V.: An automatic system for helping health consumers to understand medical texts. pp. 622–627 (2015)
2. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Inc., New York, NY, USA (1995)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
4. Brunato, D., Cimino, A., Dell'Orletta, F., Venturi, G.: 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. pp. 351–361. Association for Computational Linguistics (2016)
5. Chall, J.S., Dale, E.: Readability revisited: The new Dale-Chall readability formula. Brookline Books (1995)
6. Chiavetta, F., Lo Bosco, G., Pilato, G.: A lexicon-based approach for sentiment classification of amazon books reviews in italian language. vol. 2, pp. 159–170 (2016)
7. Chiavetta, F., Lo Bosco, G., Pilato, G.: A layered architecture for sentiment classification of products reviews in italian language. In: Monfort, V., Krempels, K.H., Majchrzak, T.A., Traverso, P. (eds.) Web Information Systems and Technologies. pp. 120–141. Springer International Publishing, Cham (2017)
8. Dell'Orletta, F., Montemagni, S., Venturi, G.: 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. pp. 73–83. Association for Computational Linguistics (2011)
9. Franchina, V., Vacca, R.: Adaptation of flesh readability index on a bilingual text written by the same author both in italian and english languages. Linguaggi **3**, 47–49 (1986)
10. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
11. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. arXiv preprint arXiv:1802.06893 (2018)
12. Hinton, G., Srivastava, N., Swersky, K.: Neural networks for machine learning lecture 6a overview of mini-batch gradient descent (2012)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
14. Huang, J., Ling, C.X.: Using auc and accuracy in evaluating learning algorithms. IEEE Transactions on knowledge and Data Engineering **17**(3), 299–310 (2005)
15. Kincaid, J.P., Fishburne Jr, R.P., Rogers, R.L., Chissom, B.S.: Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Tech. rep., Naval Technical Training Command Millington TN Research Branch (1975)

16. Lucisano, P., Piemontese, M.E.: Gulpease: una formula per la predizione della difficoltà dei testi in lingua italiana. Scuola e città **3**(31), 110–124 (1988)
17. Marschark, M., Spencer, P.E.: The Oxford handbook of deaf studies, language, and education, vol. 2. Oxford University Press (2010)
18. OECD: Inchiesta sulle competenze degli adulti primi risultati (2013)
19. Paul, P.V.: Language and deafness. Jones & Bartlett Learning (2009)
20. Shardlow, M.: A survey of automated text simplification. International Journal of Advanced Computer Science and Applications **4**(1), 58–70 (2014)
21. www.commoncrawl.org: