A Sentence based System for Measuring Syntax Complexity using a Recurrent Deep Neural Network

Giosué Lo Bosco[1], Giovanni Pilato[2], and Daniele Schicchi[1]

[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. In this paper we present a deep neural network model capable of inducing the rules that identify the syntax complexity of an Italian sentence. Our system, beyond the ability of choosing if a sentence needs of simplification, gives a score that represent the confidence of the model during the process of decision making which could be representative of the sentence complexity. Experiments have been carried out on one public corpus created specifically for the problem of text-simplification.

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

1 Introduction

Text Simplification (TS) is Natural Language process that aims at making a text more easily understandable for a determined target of people by changing the *lexical* and *syntactic* content of the original text.
The usefulness of TS can be appreciated by different kind of people, such as those who are not mother tongue or have language disabilities. For example, people affected by aphasia during the reading process have difficulties to understand syntactic structure [14], deaf children have trouble comprehending syntactically complex sentences [15] and people affected by dyslexia have comprehension difficulties in reading infrequent and long words.
For what concerns the Italian language, TS is an underdeveloped research area and this is evident from the availability of few resources and the number of developed methodologies. A cause for this is probably that the English Language is more widespread. Nonetheless, works have been done trying to face different NLP problems in Italian Language.[1, 3, 4].
The problem of evaluating the complexity of a document has already been tackled in the past using indexes like GulpEase [11] and Flesch-Vacca [6], which are based on the structural features of the sentence such us *the average number of syllables per word, the average number of words per sentence, the number of sentences* and *the average number of characters per words.* The problems with these

indexes are that they are not suitable to measure the sentence complexity and they do not consider other important aspects of the text complexity such as how much popular are the words in the text. Nowadays, the most common index for assessing sentence complexity is READ-IT[5]: a Support Vector Machine based system capable of measuring the text complexity taking into account many of different text features related to *Lexical, Morpho-syntactic and Syntactic Features* aspects. Another system capable of measuring sentence complexity for the Italian language is described in [10]. It is based on a Recurrent Neural Network used to measure the lexical and syntactic complexity of a sentence using as tokens only words and punctuation symbols.
In the domain of TS, words like *complex* and *simple* should be used keeping in mind that the complexity of a sentence is strictly related to a determined kind of people that could have different needs.Since the corpus we have used contains examples that represent the simplification process for different classes of readers, our simplification system is not specialized for any specific target reader. Nonetheless, the corpus is suited for the the goal of this work that is to understand the potentiality of a model based on Neural Network (NN) to classify Italian sentences using only the *part-of-speech*(PoS) tags which represent the syntactical aspects of the text.
In this paper, we give a contribution to the TS field using NN for developing a system capable of inducing the patterns which characterize the syntactic complexity of a sentence. Our system classifies the sentence in 2 classes *difficult-to-read* and *simple-to-read* and produces a score which represent the confidence of the network during the decision making process that could be interpreted as a measure of complexity of the given sentence.
The paper is organized as follow: in section 2 we will describe the system and our approach of facing the problem, in section 3 we will explain the methodology of carrying out the tests and results, in section 5 we will give conclusion.

2 Proposed Methodology

Our method is based on NN algorithms and it is able to discriminate if an Italian sentence needs to be simplified in order to be more easily understandable by different classed of target readers. Furthermore, the network gives a *score* that could be interpreted as a score of the sentence complexity and that represents the confidence of the network during the decision making.
To manage the task of understanding the sentence complexity we have chosen to use Recurrent Neural Networks (RNNs) that are a class of NN useful for analyzing sequences. In the recent past RNNs have shown their effectiveness in many different linguistic fields since it is well known that a sentence can be structured as a *sequence* of tokens such as words, punctuation symbols or part-of-speech.

2.1 Architecture and Parameters

We have evaluated a sentence as a sequence of *part of speech* tags calculated using a pre-trained version of TreeTagger[3] [13]. TreeTagger is a tool for annotating text with part-of-speech and it has been successfully used to tag many different languages such as German, English, Italian and so on. The tool is customizable and it allows the choice of different tag-set for each supported language. For the Italian language there exist two different tag-sets (Baroni[4] and Stein[5]) that we have separately used for parsing the sentences of the corpus.
Both the tag-sets contain tags that identify linguistic elements such as *adverbs, adjective, verb, noun* but they have different way to represent these linguistic categories. For instance, in the description of verbs one tag-set (Baroni) contains 17 different verb categories while the other one (Stein) contains 12 different verb categories. In total, the Baroni tag-set contains 52 different categories of *part-of-speech* tags while the Stein tag-set contains 38 different categories of *part-of-speech* tags.
Each *part-of-speech tag* obtained by TreeTagger is then coded as a vector using the *one-hot encoding* in which a *part-of-speech* tag becomes a vector full of 0*s* except for a *unique* one position in which the value is 1. Every sentence is evaluated as sequence of *one-hot encoded* vectors that are passed as input to the network that analyzes them. The complete process is shown in figure 1 and 2.
The network that we have used to tackle the problem of evaluating the complex-

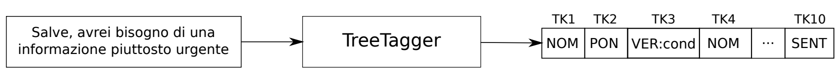

Fig. 1. Preprocessing: the sentence "Hello, I would need a rather urgent information" in Italian is evaluated as a sequence of parts of speech calculated using the TreeTagger.

ity of a Italian sentence is an RNN based on Long Short Term Memory (LSTM) artificial neurons [9]. Networks based on LSTM artificial neurons have shown good results for many sequence modeling tasks. The main features of LSTM are its abilities of facing the problem of vanishing gradient [7] and of remembering the dependencies among elements inside a sequence which are distant from each other.
The first layer of the network is made up of 512 LSTM artificial neurons. The outcome of this layer is then handled by fully connected layer composed by two neurons which use the softmax activation function. Finally, we have applied L_2 regularization [12]. The network architecture is shown in figure 2.
The probability that a sentence belongs either to a *difficult-to-read* class or a *simple-to-read* class, which is given by the last layer of the network, can be interpreted as a cumulative score that measures the complexity of the sentence by

[3] http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/
[4] http://sslmit.unibo.it/ baroni/collocazioni/itwac.tagset.txt
[5] http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/data/italian-tagset.txt

taking into account uniquely his syntactic structure.
We have used the well known *cross-entropy* as loss function which has been minimized using the RMSPROP [8] algorithm on balanced minibatch of size 50 thus each batch contains 25 *complex* sentence and 25 *simple* sentence.
To avoid overfitting, during the training process, it has been taken into account a regularization factor L_2 with a weight value of 0.01. We have limited the source sentences to 20 tokens and the network was trained for 10 epochs for both tag-sets. We have not observed any significative improvements by choosing a number of tokens greater than 20. The whole set of network parameters have been obtained through a set of trials.

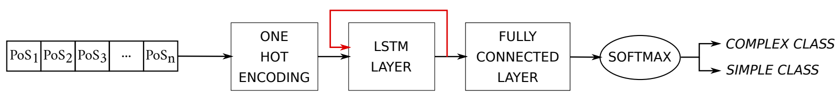

Fig. 2. Model architecture. A sentence s is structured as a sequence of *parts-of-speech* tags . Each *part-of-speech* tag is then represented as a vector through *one-hot encoding* representation.

3 Experiments and Results

3.1 Corpus

There is a lack of corpora useful to tackle the text simplification problem for the Italian language by means of machine learning algorithms. Thus we have chosen, to the best of our knowledge, the biggest available dataset created for the italian text simplification nowadays [2].
The corpus [2] contains about 63.000 pairs of sentences in which, for each original sentence, there is another corresponding sentence that keeps the same meaning and represents the simplified version of the original one. The paired sentences containing structural transformations that identify how to simplify a sentence, thus all the simplified sentences can be considered *easy-to-read* and can be used as a developmental resource for training a sentence classification algorithm.
Some of simplification rules inside the corpus are, for example, *deletion* of some words from a source sentence, *lexical substitution* of the source words so as to have a simpler sentence to understand, *insertion* of other words that can help to understand better the meaning of the sentence and so on.
The corpus has been entirely tagged with the Treetagger parser, both training and tests are based only on the tags without taking into account neither lemmas or punctuation symbols. The experiments suggest that a NN is capable of discovering the syntactics rules which characterize both approaches by understanding how to associate each sentence to the correct class.

3.2 Experiments

The evaluation of the model has proceeded using the K-FOLD cross-validation (K-FOLD) method. K-FOLD is a validation method useful for assessing the abilities of a statistical model especially in presence of few data, which is our case. In fact, 63.000 pairs of sentences are not enough to evaluate this kind of model and the use of K-FOLD evaluation methodology is necessary for clearly understand how well the classifier is capable of generalizing his knowledge to an independent dataset. The method partitions randomly the dataset into K equal sized subsets (in our case K=10): the method selects all possible K-1 subsets that are used to train the model and use the last one to validate it. The K models have been trained to classify two classes of sentences that are present into the corpus: *difficult-to-read (positive class)*, *simple-to-read (negative class)*.
The quantization of the results has been done using the Precision, Recall, True Positive Ratio (TPR) and True Negative Ratio (TNR) measures for each iteration of K-FOLD. The Recall and Precision measures, respectively, the percentage of positive class elements that the model is able to correctly classify and the percentage of mistakes that it has done during the classification of the positive class elements. TPR[6] and TNR measure respectively the proportion of elements correctly identified as positive and the proportion of elements correctly identified as negative. Finally, the results have been averaged on the K executed iterations.
We have decided to use as baseline model a support vector machine (SVM) model trained using two different kernel methods: *RBF* and *polynomial*. This choice is justified by the fact that, to our knowlege, does not exist another classification system that take as input only the *part-of-speech* tags. READ-IT can measure the syntactical complexity of a sentence but it makes available an online interface that is not handy to make a huge amount of tests. The SVM model takes as input the *part-of-speech* tags of the input sentence as a vector in which each position represents a different *part-of-speech* tag whose value is the number of the corresponding *part-of-speech* in the source text. Table 1 shows the results obtained by both models.

Model	Kernel	TAG-SET	Recall	Precision	True Positive Ratio	True Negative Ratio
RNN-S	-	STEIN	**0.819**	0.834	**0.819**	0.837
RNN-B	-	BARONI	0.764	**0.845**	0.764	0.859
SVM-SP	polynomial	STEIN	0.589	0.832	0.589	0.881
SVM-SR	RBF	STEIN	0.750	0.798	0.750	0.810
SVM-BP	polynomial	BARONI	0.506	0.839	0.506	**0.903**
SVM-BR	RBF	BARONI	0.731	0.793	0.731	0.809

Table 1. Average results of Recall, Precision, True Positive Rate, True Negative Rate for each model using both the tag-sets. We outline in bold the best value for each measure.

[6] TPR is calculated at the same way of RECALL

4 Discussion

The results show the performance of the NN model compared to those obtained by the SVM using different kernel methods. The RNN reaches the best result of Recall and, obviously, on the True Positive Ratio with the STEIN tag-set and the best result of Precision using the BARONI tag-sets. The True Negative Ratio is better using the SVM model with the polinomial kernel for both the tag-set. Although the good performance of the SVM-BP measured by the True Negative Ratio, the relative Recall measure reaches only a value of 0.506.
In our opinion, the best model is the RNN-B one that uses the BARONI tag-set, because it shows a good value of Recall that is better than the ones obtained by the SVM and the best value of Precision. Furthermore, both Recall and True Negative Ratio measures are not much different from the best ones obtained respectively by RNN-S and SVM-BP (approximately 0.05 points of difference).
The results suggest the effectiveness of our model to evaluate the syntactical complexity aspects of an Italian Sentence. The SVM model reaches a high value of True Negative Ratio that will be studied in our future works trying to understand what is the key of this outcome and if it can be embedded in the RNN-B model.
Looking into how the tag-sets influence the results we observe that both of them allow the models to obtain good value of Precision and True Negative Ratio, in fact the maximum difference, carried out as the best value minus the worst value, among the precision results is 0.052 and the maximum difference among the True Negative Results is 0.094. Conversely, their usage affect more the Recall measure in which the maximum difference is 0.313. The problem is specifically related to the polynomial kernel that seems to have more difficult to infer a considerable number of rules that identify the elements of the class *difficult-to-read*.
The good performance achieved by the models, except for the Recall of the SVM model with polynomial kernel, suggests that both tag-sets express well the syntactic features of the text and they are suited to address this kind of problem coupled with a neural model.

5 Discussion and Conclusion

We have presented a system for measuring the syntactic complexity of a sentence written in Italian language. Our system takes as input a sentence and it expresses the syntax of the sentence as a sequence of *part-of-speech* tags. The RNN at the base of our system, after learning the patterns that determine the syntactic complexity through a specific corpus created for TS, is capable of classifying a sentence as being *difficult-to-read* or *simple-to-read*. We have tested the system using two different tag-sets and we have compared the RNN with a SVM model using different kernel methods. Results show the effectiveness of the Neural Network model to address the task of classifying Italian sentences based on their readability complexity. The system can be used either as a stand-alone system or as a support tool for the creation of a complex system to address different problems such as the generation of simplified 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. 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)
3. 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)
4. 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)
5. 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)
6. 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)
7. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
8. Hinton, G., Srivastava, N., Swersky, K.: Neural networks for machine learning lecture 6a overview of mini-batch gradient descent (2012)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
10. Lo Bosco, G., Pilato, G., Schicchi, D.: A recurrent deep neural network model to measure sentence complexity for the italian language. In: Proceedings of the sixth International Workshop on Artificial Intelligence and Cognition. (2018)
11. 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)
12. Ng, A.Y.: Feature selection, l1 vs. l2 regularization, and rotational invariance. In: Proceedings of the Twenty-first International Conference on Machine Learning. pp. 78–
13. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: New methods in language processing. p. 154 (2013)
14. Shewan, C.M., Canter, G.J.: Effects of vocabulary, syntax, and sentence length on auditory comprehension in aphasic patients. Cortex **7**(3), 209 – 226 (1971)
15. Siddharthan, A.: A survey of research on text simplification. ITL-International Journal of Applied Linguistics **165**(2), 259–298 (2014)