Deep Neural Attention-Based Model for the Evaluation of Italian Sentences Complexity

Daniele Schicchi
Dipartimento di Matematica e Informatica
Università degli Studi di Palermo
Palermo, Italy

Giovanni Pilato
ICAR-CNR
National Research Council of Italy
Palermo, Italy

Giosué Lo Bosco
Dipartimento di Matematica e Informatica
Università degli Studi di Palermo
Palermo, Italy

***Abstract*—In this paper, the Automatic Text Complexity Evaluation problem is modeled as a binary classification task tackled by a Neural Network based system. It exploits Recurrent Neural Units and the Attention mechanism to measure the complexity of sentences written in the Italian language. An accurate test phase has been carried out, and the system has been compared with state-of-art tools that tackle the same problem. The computed performances proof the model suitability to evaluate sentence complexity improving the results achieved by other state-of-the-art systems.**

I. Introduction

Text Complexity Evaluation (TCE) is the process of analyzing a text to measure its grade of comprehensibility. It may involve the study of *lexical*, *syntactical* and *morphological* features, and it can be applied to many types of texts such as documents and sentences.

TCE has been widely used in many different contexts in which the measurement of text complexity is an important task. The educational environment needs tools capable of supporting the drafting of teaching material that could meet people's necessities such us those of deaf students who have to face linguistic problems arisen in their youth [1], who is affected by dyslexia that can not understand texts composed by infrequent and long words [2], or people with aphasia that have difficulties to understand syntactically complex sentences [2]. Another related application field in which Automatic Text Complexity Evaluation (ATCE) systems can affect the advancement is the Automatic Text Simplification (ATS). ATCE systems can enrich the functionality of ATS systems, or they can help as a testing tool to evaluate if the developed system creates simplification effectively.

The automation of the TCE process is a difficult problem that has been tackled by using many different methodologies. In this paper, it is presented a Neural Network (NN) model that analyzes *words* and *puntuation* symbols to evaluate the complexity of a *sentence* concerning people with low literacy skills or with soft language disabilities. Although many ATCE systems tackle the problem by analyzing documents, the creation of a methodology capable of measuring sentence complexity represents a different relevant approach.

The problem is modeled as a binary classification task, and the described methodology exploits a large public corpus released specifically for the problem of learning the factors that characterize text complexity about the language skills of the reader. The system presented in this paper has been developed and widely tested by comparing its performances with the Vec2Read system based on a similar architecture and other state-of-art classification tools that tackle the same problem.The paper is structured as follows: in section II the bibliography related to ATCE problem with a focus on systems that tackles similar problems is given, in section III the novel methodology to evaluates sentence complexity is illustrated, in section IV the corpus and the performance evaluation procedures are described, in section V and section VI a discussion on the tackled problem and conclusions are given.

II. Related Works

The need for automatic methodology capable of evaluating text complexity is a historical problem. In [3], a study whose goal was to limit the understanding difficulties encountered by enlisted personnel of the Navy was presented. It proposed a state of art model created by recomputing the numerical coefficient of the original Flesch formula based on the linguistic proficiency of the Navy subjects. For what concerns the Italian language, historical measures are a reworking of Flesch and the the GULPEASE formula [4]. More recent systems are based on Machine Learning (ML) algorithms, which are capable of inferring the factors that affect the text complexity directly from data. In [5] Support Vector Machine (SVM) model is trained on *lexical*, *syntactical* and *psycholinguistic* features extracted from Simple English Wikipedia-Wikipedia corpus in order to measure the sentence complexity. In [6] binary classification of *hard* and *simple* sentences have been carried out by means of Stochastic Gradient Descent (SGD) classifier. The training procedure exploits massAlign automatic alignment system [7] to create a corpus by aligning articles of Newsela [8].

One of the most common *sentence* complexity evaluation system for Italian language is READ-IT [9]. It relies on the SVM model trained using *La repubblica* and *Due Parole*, respectively, a newspaper considered hard for 70% of the Italian population and a newspaper which addresses low-literacy skills people written by professional linguistics. The system is characterized by a demanding pre-processing phase in which *lexical*, *syntactic*, *raw*, and *morpho-linguistic* features

are extracted, the whole set of them is then used to train the SVM model. The system outcome is the probability that the input sentence belongs to the *hard* class.
The authors of this paper have looked into the functioning of NN models to tackle different problems related to ATCE. In [10] a RNN model classifies Italian sentences considering *lexical* and *syntactical* difficulty. The model has been trained by exploiting only *words* and *punctuation symbols* and then compared with the READ-IT system. In [11], [12], the RNN model is utilized to evaluate the syntactical complexity of sentences written in Italian and English language. In [13], the problem of ATCE has been modeled as a multi-class classification task to associate sentences to more than two classes of complexity by exploiting a system based on an RNN model. Finally, a specific study of the ATCE problem character is presented in [14]. The paper carries out a set of experiments to discover how the representation of text elements affects the performance of an RNN-based model showing that the model is little affected by the changing of the representation method.

III. Methodology

In this section, a methodology capable of tackling the ATCE problem for Italian sentences is described. It models the problem as a classification task, exploiting a specific Recurrent Deep Neural Network (RNN) architecture and an efficient embedding tool to assign sentences to two different classes: *hard* (positive class) and *easy* (negative class). The methodology has been developed and widely tested. The system is divided into two modules: *pre-processing* and *core-model*, which respectively deal with the transformation of the input sentences to be evaluated by the system and the classification task. In the following, the details of the *pre-processing* module (section III-A) and the explanation of the *core-model* (section III-B) are given.

A. Preprocessing

This phase is used by many NLP systems based on ML approaches, which generally need *transforming* text elements into a numerical form that can be interpretable by the model. An effective way of representing the input text, preserving the same amount of original information, is to consider the text as a sequence composed by tokens (e.g., words, words and punctuation symbols, non-stopping words). The way to extract tokens is related to the problem domain, and it affects the system performance since the input representation must contain as much as possible useful information to accomplish the system goal. Subsequently to the *tokenization* step, every token is represented as a vector of real numbers which are suitable for the analysis by the ML model.
In this specific case, the input sentence is structured as a sequence of tokens chosen as *words* and *punctuation symbols* since the system is focused on the evaluation of *lexical* and *syntactical* features. Every token is mapped to a 300-dimensional real numbers vectors by using the dictionary resource [15], whose creation relies on Fast-Text [15], a library for efficient learning of word representation, trained on Wikipedia[1] and Common-Crawl[2]. The output of the pre-processing phase is a set of vectors organized as a matrix, which reflects the sequence of *words* and *punctuation symbols*.

B. Core Model

The system core model is composed by a stack of NN layers. The pre-processing outcome $x = (x_1, x_2, \ldots, x_T)$ is the input of the first layer composed by 512 Long Short Term Memory (LSTM) [16] units, which examines the representation matrix outputting a series of fixed-size vectors of real numbers. In details, for each row of the matrix $x_i,\ i = 1, 2, \ldots T$, 512-dimensional vector is generated taking into account the elements already analyzed. The last RNN output incorporates both *lexical* and *syntactical* content of the sentence since it is related to the sequence elements and structure. Subsequently, it is applied an *attention* layer [17] which exploits all the sequence vectors h_t generated by the LSTM layer at each timestep $t = 1, 2, \ldots, T$. The *attention* mechanism allows the model discovering the relevance of sequence elements in order to accomplish the prediction task. The output of the attention layer, called context-vector, is given by the weighted sum of the *attention weights* α_t and the output vectors h_t, $t = 1, 2, \ldots T$. The context-vector c is then passed to a dense layer function which gives the probability that the input sentence belongs to either *hard* or *easy* class. Details of the model can be found in section III-C

C. Parameters

To compute the neural network parameters which maximize the performance of the model, many *loss-function* and *optimizer* algorithms have been tried. Empirical tests show that an efficient solution is given by using the *categorical-crossentropy* [16] loss function minimized by the *RMSprop* algorithm [18].
For what concerns the model architecture, the first layer is composed of 512 LSTM neural units whose outcomes are analyzed by the attention layer activated following the mathematical formulation presented in section III-B.
The attention layer output becomes the input of the successive 2-units dense layer activated by softmax function and normalized by using a L_2 norm with a factor scale of 0.05.
The training phase has been carried out using the Early-stopping approach with a threshold of 0.001. The best results are achieved between 6 and 8 epochs.

IV. Experiments and Results

A. Corpus

The development of TS/ATCE systems for the Italian language by using ML algorithms is a new research area; this is reflected by the lacking of resources, which makes difficult the training and testing ML-based algorithms. To the best of our knowledge, the most prominent sentence-based corpus for the Italian language is the PACCSS-IT corpus [19]. It has

[1] www.wikipedia.org
[2] www.commoncrawl.org

been created by using a semi-automatic methodology whose results have been deeply assessed with the aid of human specialists. The corpus contains about 63.000 sentences hard to understand for people with low literacy skills or with soft language disabilities in which each sentence is paired with a simplified version more easily understandable by the reader.
The resource is enriched by adding the Italian sentence of *Vikidia*[3] siteVikidia follows the idea at the base of Simple-Wikipedia[4] but using different languages such as Italian. In detail, the resource contains Italian simplified version of Wikipedia articles that are coupled with the original ones. These articles are divided into sentences and then added to PACCSS-IT, considering all the Wikipedia sentences as *hard* and the Vikidia ones as *easy*. The final corpus comprise about 66.000 *easy* sentences and 90.000 *hard* sentences.

B. Vec2Read

The system has been widely tested, and it has been compared with *Vec2Read* [20] that, to the best of our knowledge, is the only one system based on *attention* mechanism which tackles the ATCE problem for the Italian language.
Vec2Read is a recently developed system based on *multi-attention* mechanism whose goal is to classify *documents* in different languages on the basis of their difficulty. Vec2Read is trained on many different English corpora, and it has been tested on different languages keeping the computed parameters. Experiments show the ability of the model to classify documents without language-specific tuning, which confirms the versatility of these classes of NN models to tackle the problem of ATCE for different languages. Indeed, a similar experiment has been shown in [11], [12] in which the same RNN-based system is capable of evaluating the syntactical complexity of sentences written in Italian or English languages.
Unfortunately, the results presented in the *Vec2Read* description paper are not reproducible at this moment since it is only available the code of the system, but neither the trained weights nor the resources used to train it are publicly available.

C. Experiments

The testing phase has been carried out by using the well-known cross-validation approach K-FOLD. Both systems, the one described in this paper and the Vec2Read, have been trained and tested on data organized using the K-FOLD mechanism. Since the data-set is unbalanced, experiments have been run three times by randomly choosing the same number of elements of both classes to make the data-set balanced. For each run, a K-FOLD division has been carried out, and final results have been averaged taking into account partial measures computed for each fold.
To make *Vec2Read* capable of evaluating the complexity of a sentence, the task has been managed as a complexity evaluation problem of documents composed by only one sentence. The system has been used by keeping the configuration presented in [20] trained for 100 epochs.

[3]vikidia.org
[4]simple.wikipedia.org

TABLE I
SYSTEMS PERFORMANCES COMPARISON. THE NAME *Attention System* INDICATES THE SINGLE-ATTENTION BASED MODEL INTRODUCED IN THIS PAPER.

Model	Epoch	Accuracy	Recall	Precision	F1-Score
Attention System	6	.880	.879	.881	.880
Vec2Read	2	.614	0.739	.599	.662

TABLE II
COMPARISON OF SYSTEMS PERFORMANCES. THE NAME *Attention System* INDICATES THE SINGLE-ATTENTION BASED MODEL INTRODUCED IN THIS PAPER.

Model	Epoch	Accuracy	Recall	Precision	F1-Score
Attention System	8	.880	.872	.887	.879
ARNN	8	.850	.830	.860	.844

To evaluate the performance of the evaluated systems, Recall, Precision, Accuracy, and F1-Score measures have been computed. Table I shows the performance in which the systems achieve the best F1-Score value.

D. Other systems

In [10], it has been presented a system based on RNN, from now on called ARNN, which, exploiting *words* and *punctuation symbols* is capable of evaluating the complexity of a sentence. The network is composed of a 512-units LSTM layer followed by a 2-units dense layer activated by softmax. The system has been trained on the original version of the PACCSS-IT and tested by using a cross-validation approach K-FOLD with K = 10. Its performances show reliability in deciding how to associate the input sentences to complexity classes, and it has been compared to READ-IT.The system presented in this paper has been compared with the ARNN to understand if an attention mechanism can improve the performance of a NN model in tackling the ATCE problem. To make the test fair, the attention-based system is trained on the original version of the PACCSS-IT corpus for a variable number of epochs by using *early-stopping* approach (see section III). The performance evaluation is carried out by using the K-FOLD methodology, with K = 10, and they are quantified using Accuracy, Recall, Precision, and F1-Score, table II show results achieved by both systems. The attention-based model overcomes the ARNN model after 2 epochs, and it achieves the best F1-Score value after 8 epochs.

V. DISCUSSION

The presented RNN system based on the attention model shows high performances in classifying sentences based on their difficulty. The system has been compared with Vec2Read [20], which has been trained for several epochs in the range of 1-100, and its performance has been measured after each epoch. Vec2Read achieves its best, evaluated as the higher value of F1-Score measure, after 2 epochs. The measures analysis makes it possible to deduce a mediocre attitude of the system to tackle the classification task. On the contrary, the presented system represents a better solution for the problem of achieving higher values of Accuracy, Recall, and

Precision. Furthermore, the presented system is built on an architecture which, compared to Vec2Read, is simpler. It exploits only one attention layer, and it has a less number of parameters to train. Moreover, the classification task is carried out by using only *words* and *punctation symbols* in contrast to Vec2Read that takes into account also *parts-of-speech* and *morphological* elements. The poor performances of Vec2Read can be explained considering that it has been created to tackle the ATCE problem for *documents*. This confirms that the problem of evaluating the sentence complexity is different from that of evaluating a document complexity. Indeed, running the Vec2Read system on document formed by only one sentence is not enough to obtain a good measure of sentence complexity. This leads to the consideration that a methodology specifically created for the analysis of many features (e.g., *lexical*, *syntactical*, *morphological*) that affect sentence complexity has to be considered.

In Section IV-D, we focused on our system, named ARNN[10], that has shown to have performances similar to those of READ-IT[9]. Table II shows that the attention-based model presented in this paper achieves the best results by using the same data for both systems. Our attention-based model significantly overcomes the results of the ARNN model after a training of 8 epochs. Since the architecture of both NN-based systems is similar, the test allows deducing that the highest results are due to the complexity of the neural model and not on the representation of data, agreeing with results shown in [14].

VI. Conclusion

A Neural Network system capable of evaluating the complexity of sentences written in the Italian language has been presented. The problem, modeled as a binary classification task, has been tackled using a system that relies on recurrent, attention, and dense layers. Although the automation of text complexity evaluation is a difficult problem, the system reaches good performances, which are computed by a cross-validation approach and measured using standard metrics. The system has been compared with state-of-art systems which tackle the same problem showing better performance on the accomplishment of the evaluation task. Future works will consider studies about how sarcasm can affect text complexity [21], an extension of the neural model, its comparison with other techniques such as [22] and trying to exploit them in the robotics domain [23].

Acknowledgment

This research has been partially supported by AMICO Project, CUP B46G18000390005; cod ARS01 00900 "Assistenza Medicale In COntextual awareness" decreto di concessione del 10 luglio 2018 prot. n.11598.

References

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

[2] A. Siddharthan, "A survey of research on text simplification," *ITL-International Journal of Applied Linguistics*, vol. 165, no. 2, pp. 259–298, 2014.

[3] J. Kincaid, *Derivation of New Readability Formulas: (automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel*, ser. Research Branch report. Chief of Naval Technical Training, Naval Air Station Memphis, 1975.

[4] P. Lucisano and M. E. Piemontese, "Gulpease: una formula per la predizione della difficoltà dei testi in lingua italiana," *Scuola e città*, vol. 3, no. 31, pp. 110–124, 1988.

[5] S. Vajjala and D. Meurers, "Assessing the relative reading level of sentence pairs for text simplification," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 2014, pp. 288–297.

[6] C. Scarton, G. Paetzold, and L. Specia, "Text simplification from professionally produced corpora," in *Proceedings of the LREC-2018*. Miyazaki, Japan: European Languages Resources Association (ELRA), May 2018.

[7] G. Paetzold, F. Alva-Manchego, and L. Specia, "Massalign: Alignment and annotation of comparable documents," in *Proceedings of the IJCNLP 2017, System Demonstrations*, 2017, pp. 1–4.

[8] W. Xu, C. Callison-Burch, and C. Napoles, "Problems in current text simplification research: New data can help," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 283–297, 2015.

[9] F. Dell'Orletta, S. Montemagni, and G. Venturi, "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, 2011, pp. 73–83.

[10] G. Lo Bosco, G. Pilato, and D. Schicchi, "A recurrent deep neural network model to measure sentence complexity for the italian language," in *Proceedings of the AIC 2018*, vol. 2418. CEUR-WS, 2019, pp. 90–97.

[11] G. Lo Bosco, G. Pilato, and D. Schicchi, "A sentence based system for measuring syntax complexity using a recurrent deep neural network." in *2nd Workshop on Natural Language for Artificial Intelligence, NL4AI 2018 at AI* IA*, 2018, pp. 95–101.

[12] D. Schicchi, G. Lo Bosco, and G. Pilato, "Machine learning models for measuring syntax complexity of english text," in *Biologically Inspired Cognitive Architectures 2019*. Cham: Springer International Publishing, 2020, pp. 449–454.

[13] A. Cuzzocrea, G. Lo Bosco, G. Pilato, and D. Schicchi, "Multi-class text complexity evaluation via deep neural networks," in *Proc. of IDEAL 2019*. Springer, 2019, pp. 313–322.

[14] G. Lo Bosco, G. Pilato, and D. Schicchi, "A neural network model for the evaluation of text complexity in italian language: a representation point of view," *Procedia Computer Science*, vol. 145, pp. 464 – 470, 2018.

[15] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," *CoRR*, vol. abs/1802.06893, 2018.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[17] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Sep. 2017, pp. 1615–1625.

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

[19] D. Brunato, A. Cimino, F. Dell'Orletta, and G. Venturi, "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*, Nov. 2016, pp. 351–361.

[20] I. Madrazo Azpiazu and M. S. Pera, "Multiattentive recurrent neural network architecture for multilingual readability assessment," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 421–436, 2019.

[21] M. Di Gangi, G. Lo Bosco, and G. Pilato, "Effectiveness of data-driven induction of semantic spaces and traditional classifiers for sarcasm detection," *Natural Language Engineering*, vol. 25, no. 2, pp. 257–285, 2019.

[22] G. Pilato and G. Vassallo, "TSVD as a statistical estimator in the latent semantic analysis paradigm," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 2, pp. 185–192, 2014.

[23] A. Chella, R. E. Barone, G. Pilato, and R. Sorbello, "An emotional storyteller robot." in *AAAI Spring Symposium: Emotion, Personality, and Social Behavior*, 2008, pp. 17–22.