Machine learning models for Measuring Syntax Complexity of English Text

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

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

Abstract. In this paper we propose a methodology to assess the syntax complexity of a sentence representing it as sequence of parts-of-speech and comparing Recurrent Neural Networks and Support Vector Machine. We have carried out experiments in English language which are compared with previous results obtained for the Italian one.

Keywords: text-simplification, deep-learning, machine-learning

1 Introduction

Natural Language Processing (NLP) is a research area that tackles the problem of analyzing in a automated manner natural language data. Researchers who work in NLP field have created many interesting models capable of solving problems, for example, related to computational creativity [1], teaching [2], machine translation [3], support system [4] and so on.
Text Simplification (TS) is a branch of NLP that aims at making a text more easily understandable for people. A core part of TS system is the *evaluation* process that, taking into account both reader skills and text complexity, decides if the text needs of being simplified. The evaluation of text complexity (TE) is not a trivial problem and it is an actual research topic since the performances of TS system are connected to this task in many ways. Furthermore, TE system can be used both as support for TS and as independent system. For example, it can be appreciated as *decision support system* by people in contact with different communities such as those who are not mother tongue or have language disabilities.
An historical measure of text complexity is the Flesch–Kincaid [5] index which is based on structural features of the text and it gives a degree of complexity evaluating *total words*, *total sentences*, *total syllables*. However, it is a common opinion that the evaluation of only structural features is not representative of total text complexity. In the recent years, more reliable indexes were developed. They express the degree of text complexity considering more sophisticated features like the frequency of words and *simple* word dictionary, the depth of parse tree and text morphology. READ-IT [6] is a Support Vector Machine based system created to tackle the problem of TE. The system takes into account

This is a pre-copyedited version of a contribution published in Biologically Inspired Cognitive Architectures 2019, BICA 2019, Advances in Intelligent Systems and Computing, vol 948. Edited by Alexei V. Samsonovich, published by Springer, Cham. The definitive authenticated version is available online via https://doi.org/10.1007/978-3-030-25719-4_59.

Lexical, *Morpho-syntactic* and *Syntactic* features aspects to decide what category of complexity belongs the input text. In order to evaluate the performance of TS system it has been proposed FKBLEU [7] the SARI index [7] . The authors proposed a TE data-driven system based on Neural Network (NN) which measure the complexity of Italian sentences taking into account *lexical* and *syntactical* aspects [8].
In this paper it is presented a TE system whose objective is focused on the evaluation of *syntax* complexity of sentences in English Language. The paper is organized as follow: in section 2 we describe components of the system, in section 3 we explain the way of evaluating the system performance, in section 4 we will give the conclusions.

2 Proposed Methodology

The purpose of the system is to understand rules that identify *syntactical* constructs which make a sentence hard to understand for a reader. We have evaluated the performance of two different machine learning algorithms: the Recurrent Neural Networks (RNN) and the Support Vector Machine (SVM).
The RNN is a powerful model created for the elaboration of data sequence that can be used for the NLP field if the input text is structured as a sequence of tokens. The SVM [9] is a ML algorithm that has been widely used to solve different kind of problems. It has already used for NLP problems [6] showing great potentiality to analyze texts. For both models the input sentence is preprocessed by a module that extract its *parts-of-speech* and that makes it suitable for the analysis.

2.1 Preprocessing

The preprocessing module serves to represent the sentence as sequence of *part-of-speech* and to make it suitable for the analysis from the ML models. The identification of the *parts-of-speech* is carried out using a pre-trained version of TreeTagger [10]. TreeTagger is a tool capable of annotating text with its *parts-of-speech* in different languages such as Italian, English, German and so on. It allows tagging different languages by means of parameter files, called *tagsets*, that include the instructions to extract *parts-of-speech* from a text. We have used the *BNC tagset*[3] which allows to draw out 61 different *parts-of-speech* belonging to different categories like *verbs*, *adverbs*, *punctuation* and *pronouns*.
After the extraction process, it is applied a transformation that identifies each element as a vector of real numbers using the well known *one-hot encoding*. Using the *one-hot encoding* every sentence is represented by a sequence of vectors in which each of them identifies uniquely a *part-of-speech*.
The RNN model us compared with a Support Vector Machine (SVM). Since the SVM is not suitable to examine sequence of vectors the sentence preprocessing

[3] http://www.natcorp.ox.ac.uk/docs/c5spec.html

is slightly different. In this case a sequence is represented by a single vector of length equal to the total amount of *parts-of-speech* and each position of the vector identifies the occurrence of a specific *part-of-speech* in the sequence. After the counting, we have normalized the vector by the total number of *parts-of-speech*, in this specific case 61.

2.2 Architectures and Parameters

The RNN model is based on Long Short Term Memory (LSTM) [11] artificial neurons which have shown good performance tackling problems belonging to NLP field related to sequence modeling tasks. The architecture of the Network is composed by 3 layers. The first is the *input layer* whose job is to pick data after the *preprocessing* phase and making it accessible to the *LSTM layer*. The *LSTM layer*, consisting of 512 LSTM units that is responsible for analyzing the sequence. The output of this layer stimulates the next *dense layer* which is activated using *sotmax* [12] activation function giving the probability that the sequence belongs either to *easy-to-understand* or to *hard-to-understand* class. The last level is regularized using the L_2 regularization factor with value of 0.01. The network has been trained using minibatch of size 50 divided as 25 *easy-to-understand* sentences and 25 *hard-to-understand* sentences. The set of parameters have been obtained through a set of experiments which suggest what are good configurations for solving the TE problem.
The SVM is a learning method with solid computational learning theory principles behind its functioning [9]. It has been developed by the aid of scikit-learn[4] version 0.20.2, a library that helps for the implementation of ML algorithms using different kernels methods: *linear*, *RBF* and *polynomial*.

3 Experiments and Results

3.1 Corpus

To understand the *syntactical* complexity of a sentence using a data driven approach it is needed to use a specific corpus that contains sentences labeled as *hard to understand* or *easy to understand*. Our choice fell on Newsela [13] corpus that we have used to train and test the system. The corpus is a collection of articles which have been simplified by human experts. Each article has been simplified 4 times, the original document is marked with the label 0 and its easier versions are labeled with progressive numbers, 5 identifies the most simplified. Unfortunately, the Newsela corpus does not give any information about the complexity of sentences inside documents but it provides a cumulative measure associates to the document. Thus, the document could contain sentences that do not reflects the complexity of the belonging document. The solution that we propose is to take as *hard-to-understand* all the sentences inside documents marked with $0, 1$ labels which are not present in documents with labels strictly

[4] https://scikit-learn.org

greater than 1. The *easy-to-understand* sentences are picked as all the sentences inside documents identified with labels $4, 5$ which are not found in documents with label strictly less than 4. Using this process we have harvested approximately 130.000 *hard-to-understand* and 80.000 *easy-to-understand* sentences.

3.2 Experiments and Discussion

The system has been tested using a cross-validation approach known as K-FOLD with $K = 10$. The K-FOLD method is often used for testing the performance of machine learning and It consists in the creation of a dataset partition in K sets. The training phase is iterated K times exploiting K-1 sets as training-set and the last one as validation-set which means that all the sets are used alternately as validation-set and training-set. For each iteration we have measured the well known Recall, Precision, True Negative Ratio (TNR) and True Positive Ratio (TPR) and after all the iterations we have averaged the obtained results. The table 1 shows the comparison of these two models. Since the NN has been trained for a variable number of epochs we have decided to choose the ones trained for 3 (LSTM-3) and 5 (LSTM-5) epochs for the comparison.

Model	Kernel	TAG-SET	Recall	Precision	TPR	TNR
LSTM-5	-	BNC	.826	.849	.826	.853
LSTM-3	-	BNC	.792	.873	.792	**.884**
SVM-L	Linear	BNC	.815	**.890**	.815	.834
SVM-R	RBF	BNC	.857	.825	.857	.699
SVM-P	Polynomial	BNC	**.999**	.624	**.999**	0.0

Table 1. Average results of Recall, Precision, TPR, TNR calculated according to 10-FOLD.

Results shows that both LSTM and SVM are capable of classifying quite well data of the two classes. The SVM-R reaches the best result in the classification of *hard-to-understand* sentences obtaining a good value of Recall but it often makes mistakes to classify *simple-to-understand* sequences. The LSTM-5 instead keeps a balanced behavior for the classification of sentences of both classes. Furthermore, the LSTM models are more precise than SVM-R during the process of classification of *hard-to-understand* sentences. The SVM-L is capable of reaching a good value of Recall and the maximum value of Precision. The SVM-L can be compared directly with the LSTM-3 which reaches slightly lower values of Precision and Recall but a substantial higher value of TNR. The SVM-P is the worst model since the results show highest recall but with low precision and the worst value of TNR, in this case the model is too unbalanced toward the *hard-to-understand* sentences.
Although the measures show good performance of SVM-L it should be taken into account the computational effort to create the representation vectors (section

2.1) suitable for the SVM. Instead, the RNN only need of the *one-hot encoded* vectors which cost is negligible.
We have carried out experiments training the network for a variable number of epochs from 1 to 10. The RNN reaches good performance already from the training for 1 epoch demonstrating high values of Precision, TNR and Recall. In this regard, the Recall value increases when the model is trained for more epochs. However, it results that an higher number of epochs lower the values of Precision and TNR.
This paper is a part of a series of documents that show the potentiality of Neural Networks for the evaluation of text complexity [8, 14, 15]. The architecture of the Network has already been proposed to evaluate the syntax complexity of sentences in Italian language [14]. This paper shows further proofs that the RNN is capable of tackling the problem with good results for different languages. The table 2 shows the comparison between the results associated to the RNN and the SVM for classification of Italian and English sentences.

Model	Kernel	TAG-SET	Recall	Precision	TPR	TNR
LSTM-IT-S	-	STEIN	.819	.834	.819	.837
LSTM-IT-B	-	BARONI	.764	.845	.764	.859
LSTM-EN-5	-	BNC	.826	.849	.826	.853
LSTM-EN-3	-	BNC	.792	.873	.792	.884
SVM-EN-P	Polynomial	BNC	.999	.624	.999	0.0
SVM-EN-L	Linear	BNC	.815	.890	.815	.834
SVM-EN-R	RBF	BNC	.857	.825	.857	.699
SVM-IT-SP	Polynomial	STEIN	.589	.832	.589	.881
SVM-EN-L	Linear	STEIN	.629	.768	.629	.810
SVM-IT-SR	RBF	STEIN	.750	.798	.750	.810
SVM-IT-BP	Polynomial	BARONI	.506	.839	.506	.903
SVM-EN-L	Linear	BARONI	.596	.767	.596	.819
SVM-IT-BR	RBF	BARONI	.731	.793	.731	.809

Table 2. Comparison of SVM model trained on Italian and English language based on the average of Recall, Precision, TPR, TNR.

4 Conclusions

We have presented a comparison of systems based on RNN and SVM ML algorithms for the evaluation of syntax complexity of sentences. The approach is completely data driven and it shows the abilities of Neural Network and SVM of tackling the problem in different languages, Italian and English. Experiments describe good performances of both models for the English language. On the contrary of SVM the RNN shows great versatility discovering rules that identifies the sentence complexity also for the Italian language.

References

1. Schicchi, D., Pilato, G.: Wordy: A semi-automatic methodology aimed at the creation of neologisms based on a semantic network and blending devices. In: L. Barolli, O. Terzo (eds.) Complex, Intelligent, and Software Intensive Systems, pp. 236–248. Springer International Publishing, Cham (2018)
2. Schicchi, D., Pilato, G.: A social humanoid robot as a playfellow for vocabulary enhancement. In: 2018 Second IEEE International Conference on Robotic Computing (IRC), pp. 205–208. IEEE Computer Society, Los Alamitos, CA, USA (2018)
3. Di Gangi, M.A., Federico, M.: Deep neural machine translation with weakly-recurrent units. In: 21st Annual Conference of the European Association for Machine Translation, pp. 119–128 (2018)
4. Alfano, M., Lenzitti, B., Lo Bosco, G., Perticone, V.: An automatic system for helping health consumers to understand medical texts. pp. 622–627 (2015)
5. Kincaid, J.: Derivation of New Readability Formulas: (automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel. Research Branch report. Chief of Naval Technical Training, Naval Air Station Memphis (1975)
6. 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)
7. Xu, W., Napoles, C., Pavlick, E., Chen, Q., Callison-Burch, C.: Optimizing statistical machine translation for text simplification. Transactions of the Association for Computational Linguistics **4**, 401–415 (2016). DOI 10.1162/tacl_a_00107
8. 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)
9. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)
10. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: New methods in language processing, p. 154 (2013)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
12. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
13. Xu, W., Callison-Burch, C., Napoles, C.: Problems in current text simplification research: New data can help. Transactions of the Association for Computational Linguistics **3**, 283–297 (2015). DOI 10.1162/tacl_a_00139
14. Lo Bosco, G., Pilato, G., Schicchi, D.: 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, vol. 2244, pp. 95–101. CEUR-WS (2018)
15. Bosco, G.L., Pilato, G., Schicchi, D.: A neural network model for the evaluation of text complexity in italian language: a representation point of view. Procedia computer science **145**, 464–470 (2018)