Learning to Classify Text Complexity for the Italian Language Using Support Vector Machines

Valentino Santucci[1(✉)], Luciana Forti[1], Filippo Santarelli[2], Stefania Spina[1], and Alfredo Milani[3]

[1] Department of Humanities and Social Sciences, University for Foreigners of Perugia, Perugia, Italy
{valentino.santucci,luciana.forti,stefania.spina}@unistrapg.it
[2] Istituto per le Applicazioni del Calcolo (CNR), Roma, Italy
f.santarelli@iac.cnr.it
[3] Department of Mathematics and Computer Science, University of Perugia, Perugia, Italy
alfredo.milani@unipg.it

Abstract. Natural language processing is undoubtedly one of the most active fields of research in the machine learning community. In this work we propose a supervised classification system that, given in input a text written in the Italian language, predicts its linguistic complexity in terms of a level of the Common European Framework of Reference for Languages (better known as CEFR). The system was built by considering: (i) a dataset of texts labeled by linguistic experts was collected, (ii) some vectorisation procedures which transform any text to a numerical representation, and (iii) the training of a support vector machine's model. Experiments were conducted following a statistically sound design and the experimental results show that the system is able to reach a good prediction accuracy.

Keywords: Text classification · Natural Language Processing · Support vector machines

1 Introduction

Natural Language Processing (NLP) is emerging in the recent years as one of the most researched and popular topics in the machine learning community [11,19,23]. NLP based tools allows to develop several real-world applications such as automatic translation, text summarization, speech recognition, chatbots and question answering, etc.

Another interesting application is the classification of a text in different levels of complexity [7] which is key in mood and sentiment analysis, in the detection of hate speech [18], in text simplification, and also in the assessment of text readability in relation to both native and non-native readers.

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12250, pp. 367–376, 2020.
https://doi.org/10.1007/978-3-030-58802-1_27

In this work we propose and analyze a supervised learning system for the automatic classification of a text, written in Italian, into different complexity levels according to the Common European Framework of Reference for Languages (CEFR) [8]. The proposed system is freely available online[1] and it can be used in a variety of scenarios like, for instance, to choose texts to be used in a lesson or as part of a language test.

From the computational point-of-view, the supervised system was implemented as a Support Vector Machine (SVM) [20] which learns a numerical model from a *vectorised* representation of the texts. In fact, texts are converted to numeric vectors which correspond to linguistic features computed on top of the tokens, part-of-speech tags and syntactic trees of the given texts.

Therefore, our work focuses both on the computational procedures for calculating the features and on the SVM implementation for learning a classification model.

Regarding the dataset, we have collected 692 texts in Italian language labeled by experts in the field using four levels of the CEFR. Though the dataset is not huge, a thorough tuning of the SVM parameters and features selection procedures allowed to obtain a classification system with good performances.

The rest of the paper is organized as follows. The main design of the system is presented in Sect. 2. The classification model and the numerical features are depicted in, respectively, Sects. 3 and 4. An experimental investigation is provided in Sect. 5, while Sect. 6 concludes the paper by also drawing future lines of research.

2 System Design

The task of learning to classify the complexity of a text has been approached using a supervised classification system whose design is depicted in this section.

Since it is supervised learning, training texts – already classified – are required. For this reason, we collected a dataset of texts labeled by the experts of the CVCL center of the University for Foreigners of Perugia[2]. The texts in the dataset are labeled by means of four increasing levels of difficulty. In order to be compliant with the world of linguistic certifications, the four CEFR proficiency levels B1, B2, C1 and C2 were used[3]. Clearly, this four levels are the target classes of the supervised classification system.

In total, the collected dataset is composed by 692 texts divided among the four classes as depicted in Table 1 which also provides quantitative information about the number of tokens.

[1] https://lol.unistrapg.it/malt.

[2] CVCL is the acronym of "Centro Valutazione Certificazioni Linguistiche" which can be translated to "Evaluation Center for Linguistic Certifications". Its website is reachable at https://www.cvcl.it.

[3] CEFR (Common European Framework of Reference for Languages) actually has 6 levels, but the levels A1 and A2 were omitted from our study because linguistic experts do not find them significant for this investigation.

Table 1. Characteristics of the dataset

Class	#Texts	#Tokens
B1	249	45 695
B2	185	90 133
C1	139	95 515
C2	119	104 679
Dataset	692	336 022

Though the four classes have an intrinsic order of difficulty, in this work we ignore such ordering thus to be able to rely on the most used classification models available in the literature. Nevertheless, we must stress that this is not limiting. In fact, in a preliminary investigation, described in [10], we have experimentally proved that considering the intrinsic order of the classes is not relevant for our task.

Figure 1 depicts the main design of the system.

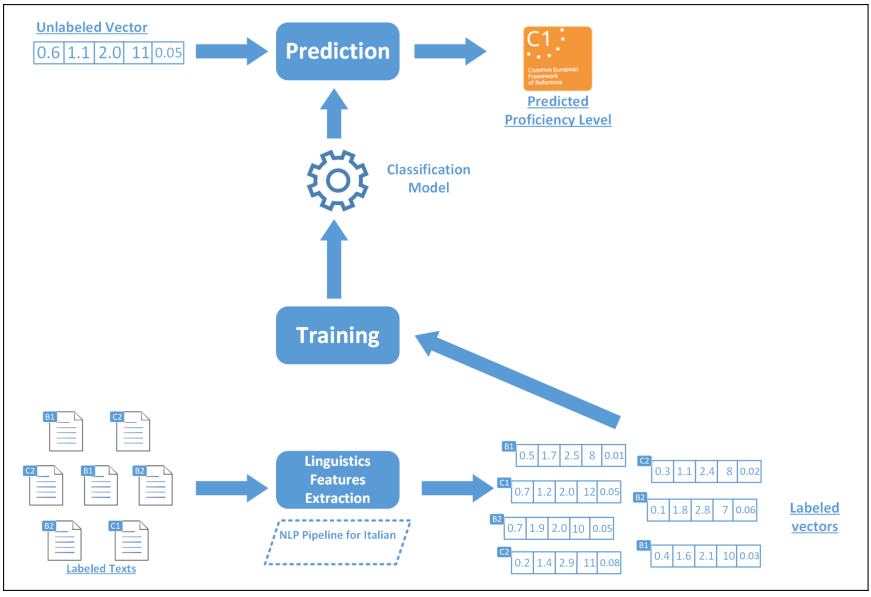

Fig. 1. Main design of the system

The classification model does not directly work with the texts in their pure form. In fact, any text is converted into a vector of numeric features so that learning and classification can employ numerical models. Such numerical vectors are obtained by computing quantitative linguistic features on top of the elaboration performed by NLP pipeline tools for the Italian language.

First, the inner parameters of the classification model are trained using the labeled vectors corresponding to the texts in the considered dataset. Then, any unlabeled text is *vectorised* and fed to the trained model which predicts its proficiency level. Interestingly, not only the predicted class is returned, but the system also provides a normalized distribution of values, one for each class, expressing how likely is the analyzed text to belong to a given class.

This architecture allows, on the one hand, to use the most common classification models available in the machine learning literature [20] and, on the other hand, to build a classification model based only on the linguistic features of the texts that, we believe, are what discriminate texts from the point-of-view of the CEFR levels.

Finally, a user friendly web interface was developed as depicted in Fig. 2: the user types or pastes a text of his/her choice in the provided text area, press the "Analyse" button, then the system transparently executes the prediction procedure of the trained model and shows the predicted CEFR level for the given text, together with a chart showing how the four different levels are represented within the text in terms of percentages. Moreover, additional charts can be recalled by using the buttons on the result page.

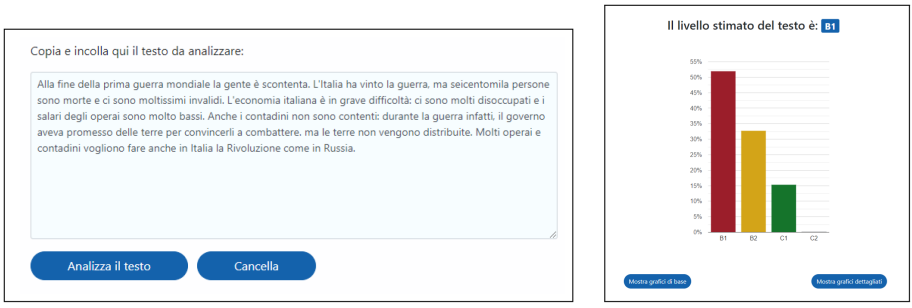

Fig. 2. User interface: the input form and the results of the elaboration

The developed resource is freely available on the web at the following address: https://lol.unistrapg.it/malt.

3 The Classification Model

Regarding the classification model, we made some preliminary experiments using decision trees, random forests, feedforward neural networks and support vector machines. Some of these experiments are described in [10,15]. According to the preliminary results, this work focuses on the Support Vector Machines (SVM) model. Interestingly, given the small size of the dataset, SVM look to work better than the trendy neural network models.

An SVM [20] is a supervised classification model which, given a (training) set of labeled numeric vectors, constructs a set of hyperplanes in a high-dimensional

space, which identify the regions of the space corresponding to the different labels, i.e., the CEFR levels in our case.

The SVM implementation of the popular *Sci-Kit Learn* library [17] has been used, while the Gaussian radial basin functions have been considered as kernel functions of the SVM.

4 Numerical Features of a Text

In order to compute the numerical linguistic features for feeding the classifier, we have used a NLP pipeline library which takes into account the Italian language. We found three libraries freely available: Tint [16], UDPipe [21] and Spacy [13]. After some preliminary experiments, we decided to proceed with UDPipe because, from our investigation, it was the most reliable for the Italian language.

The UDPipe library has been used in order to:

- tokenize a text and also split it in sentences,
- annotate any token with its lemma, its part-of-speech tag and other morpho-syntactic properties,
- parse a text in order to build dependency trees for the sentences contained in the text.

Moreover, since constituent trees were not directly computable in UDPipe, we have used the constituent parser for the Italian language of the OpenNLP project [5].

The features considered in this work can be divided in six categories:

1. raw text features,
2. lexical features,
3. morphological features,
4. morpho-syntactic features
5. discursive features,
6. syntactic features.

Some of them are formed by only one number, while others are vectors of several real numbers. Anyway, the vectors of all the numerical features are concatenated in order to form a unique real-valued vector for the given text in input. The length of such a vector is 139. Hence, after the extraction of the numerical features, any text is embedded in the space $\mathbb{R}^{139}$.

In the following, we provide the description of the calculation procedure for the some of the features considered.

4.1 Raw Text Features

The raw text features are computed after the tokenization and include statistics such the average and standard deviation of: the sentence length in tokens, the token length in characters, the text length in sentences and in lemmas.

4.2 Lexical Features

The lexical features are computed basing on the lemmatization of the tokens in the texts. They include statistics such as: the amount of lemmas in the text which are classified in order of availability in a reference vocabulary[4]; the number of nouns considered as Abstract, Semiabstract and Concrete; the lexical diversity, i.e., the ratio between the total number of words and the total number of unique words; etc.

4.3 Morphological Features

Morphological features are reflected by the Morphological Complexity Index (MCI) computed for two word classes: verbs and nouns. The MCI is operationalised by randomly drawing sub-samples of 10 forms of a word class (e.g.. verbs) from a text and computing the average within-sample and across-samples of inflectional exponents. Further details can be found in [6].

4.4 Morpho-Syntactic Features

Based on part-of-speech (POS) tagging and the morphological analysis conducted by UDPipe, statistic values about the following morpho-syntactic features are computed: the subordinate ratio, i.e., the percentage of subordinate clauses over the total number of clauses; the POS tags distribution; the verbal moods distribution; and the dependency tags distribution.

4.5 Discursive Features

Discursive features concerns the cohesive structure among the sentences in a text. In this work we have considered the referential cohesion and the deep causal cohesion.

4.6 Syntactic Features

Based on the dependency and constituent trees of the text in input, some statistics about the syntactic structure of the text are considered as follows: the depth of the parse trees, the non-verbal path lengths, the size of the maximal non-verbal phrase, the percentage of verbal roots, the arity of verbal nodes, etc.

5 Experiments

Experiments were conducted using the SVM classifier model available in the commonly used *SKLearn* module of the Python 3 programming environment [17].

[4] "Nuovo Vocabolario di Base De Mauro" that translates to "New Basic Italian Vocabulary".

Every experiment – tuning the SVM hyper-parameters, selecting the features, and assessing the final accuracy of the system – was performed using 5 repetitions of a stratified 10-folds cross-validation executed on the whole dataset of 692 texts.

First, the hyper-parameters C and γ of the SVM model have been tuned by means of a grid search process aimed at optimising the F1 score measure. This measure has been used in order to avoid issues due to the unbalanced nature of the dataset. The whole set of 139 features was considered and the calibrated setting is $C = 2.24$ and $\gamma = 0.02$.

After this tuning, a *features selection* phase was designed by using the well known Recursive Features Elimination (RFE) algorithm [12]. RFE recursively fits the model and removes the weakest feature until a specified number of features is reached. In our work, the well known *permutation feature importance* technique [9] was considered to measure the importance of every feature during the last model fitting. Moreover, to find the optimal number of features, cross-validation was used with RFE to score different feature subsets and select the best scoring collection of features. As depicted in Fig. 3, a subset formed by 54 features – around the 39% of the whole set of features – has obtained the best F1 score in our experiments.

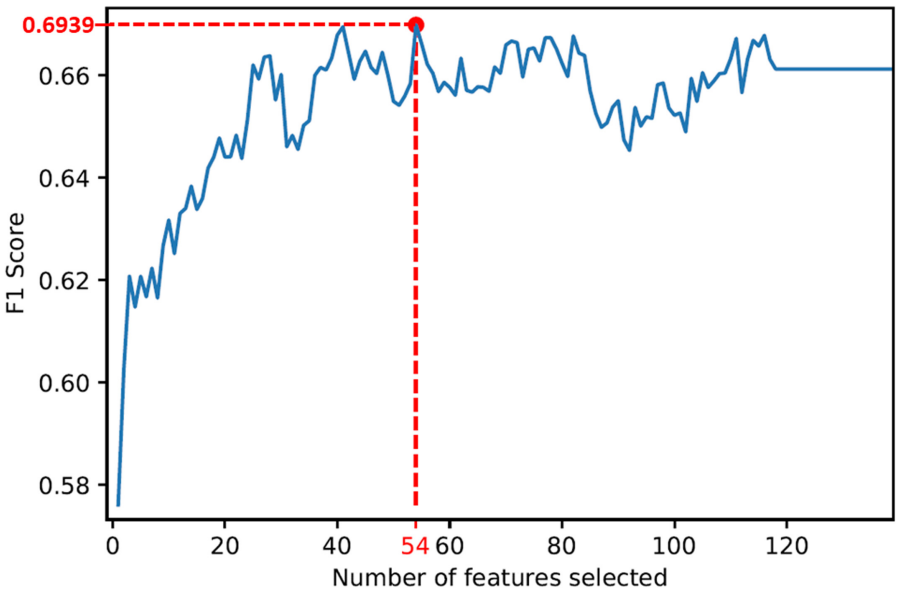

Fig. 3. Features selection graph

Then, the performances of the tuned SVM model trained using the selected features are shown in the confusion matrix provided in Table 2. In this table, each entry X, Y provides the average number – over the 5 repetitions of the

10-folds cross-validation process – of texts which are known to belong to class X, but have been classified by our system to class Y.

Table 2. Confusion matrix

Classes	B1	B2	C1	C2
B1	**214.6**	30.2	4.2	0.0
B2	28.4	**129.8**	23.8	3.0
C1	9.6	28.8	**74.2**	26.4
C2	1.2	9.0	30.0	**78.8**

The correctly classified texts are those in the diagonal of the confusion matrix. They are (in average) 497.4 out of 692, thus the accuracy of our system is about 71.88%.

The confusion matrix also allows to derive the precision and recall measures [20] for all the considered CEFR levels. Our experiments reveal that the B1 level exhibits the highest precision and recall (respectively, 84.55% and 86.18%), while the weakest predictions are those regarding the C1 level (which has 56.13% and 53.38% as, respectively, precision and recall).

Furthermore, it is interesting to observe that most of the incorrectly classified texts are only one level away from their actual CEFR levels. In fact, by aggregating the pairs of levels B1, B2 and C1,C2 into the macro-levels B and C, respectively, we obtain that the average accuracy of the system increases up to 88.50%.

Finally, note that the results discussed in this section are also in line with the 2D visualisations of the dataset provided in Fig. 4 which provides the result of two different executions of the well known dimensionality reduction technique t-SNE [22] executed on the 139-dimensional representation of the dataset. Each point is the two-dimensional representation of a text.

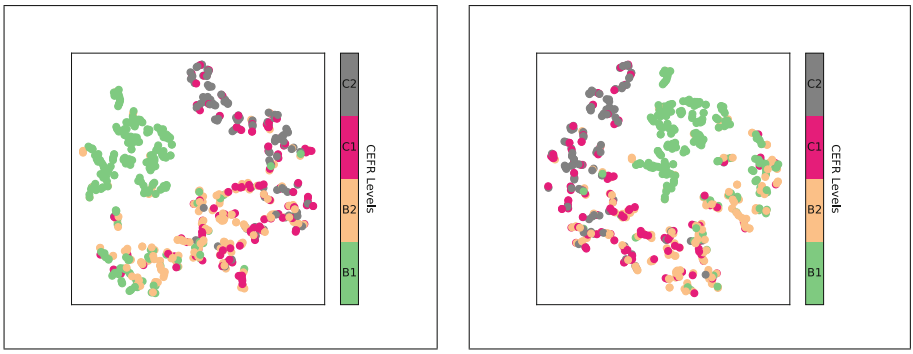

Fig. 4. 2D visualisations of the dataset obtained with two executions of t-SNE.

6 Conclusion and Future Work

In this work we have introduced an NLP tool able to automatically assess the proficiency level of an Italian text used for second language learning purposes.

A dataset of texts labeled by experts was used to learn and evaluate the performance of an SVM classifier model which is trained using linguistic features measured quantitatively and extracted from the texts.

Experiments were held in order to analyze the effectiveness and the reliability of the proposed prototypical classification system. Overall, the classification accuracy obtained is very good and satisfactory for the linguistic experts that use our tool.

Further improvement to our system can be obtained by collecting more data, i.e., more texts labeled by experts, but an interesting future line of research which, in our opinion, deserves to be deeply investigated is the automatic augmentation of the text dataset. Moreover, it can be interesting to include, in the learning procedure, algorithms from the field of evolutionary computation like, for instance, those proposed in [1–4,14].

Acknowledgments. This research was partially supported by: (i) the grant 2018.0424.021 *MALT-IT2. Una risorsa computazionale per Misurare Automaticamente la Leggibilità dei Testi per studenti di Italiano L2*, co-funded by the University for Foreigners of Perugia and by the Fondazione Cassa di Risparmio di Perugia; (ii) Università per Stranieri di Perugia – *Finanziamento per Progetti di Ricerca di Ateneo – PRA 2020*; (iii) Università per Stranieri di Perugia – Progetto di ricerca *Artificial Intelligence for Education, Social and Human Sciences*.

References

1. Baioletti, M., Milani, A., Santucci, V.: A new precedence-based ant colony optimization for permutation problems. In: Shi, Y., et al. (eds.) SEAL 2017. LNCS, vol. 10593, pp. 960–971. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68759-9_79
2. Baioletti, M., Milani, A., Santucci, V.: Learning Bayesian networks with algebraic differential evolution. In: Auger, A., Fonseca, C.M., Lourenço, N., Machado, P., Paquete, L., Whitley, D. (eds.) PPSN 2018. LNCS, vol. 11102, pp. 436–448. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99259-4_35
3. Baioletti, M., Milani, A., Santucci, V.: MOEA/DEP: an algebraic decomposition-based evolutionary algorithm for the multiobjective permutation flowshop scheduling problem. In: Liefooghe, A., López-Ibáñez, M. (eds.) EvoCOP 2018. LNCS, vol. 10782, pp. 132–145. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-77449-7_9
4. Baioletti, M., Milani, A., Santucci, V.: Variable neighborhood algebraic differential evolution: an application to the linear ordering problem with cumulative costs. Inf. Sci. **507**, 37–52 (2020). https://doi.org/10.1016/j.ins.2019.08.016
5. Baldridge, J.: The opennlp project, p. 1 (2005). http://opennlp.apache.org/index.html. Accessed 2 Feb 2012
6. Brezina, V., Pallotti, G.: Morphological complexity in written L2 texts. Second Lang. Res. **35**(1), 99–119 (2016). https://doi.org/10.1177/0267658316643125

7. 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, Edinburgh, Scotland, UK, pp. 73–83. Association for Computational Linguistics, July 2011. https://www.aclweb.org/anthology/W11-2308
8. Council of Europe, Modern Languages Division. Council for Cultural Co-operation. Education Committee. Common European Framework of Reference for Languages: learning, teaching, assessment. Cambridge University Press (2001)
9. Fisher, A., Rudin, C., Dominici, F.: Model class reliance: variable importance measures for any machine learning model class, from the "rashomon" perspective. arXiv preprint arXiv:1801.01489 (2018)
10. Forti, L., Milani, A., Piersanti, L., Santarelli, F., Santucci, V., Spina, S.: Measuring text complexity for Italian as a second language learning purposes. In: Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, Florence, Italy, pp. 360–368. Association for Computational Linguistics, August 2019. https://www.aclweb.org/anthology/W11-2308
11. Goldberg, Y.: Neural network methods for natural language processing. Synth. Lect. Hum. Lang. Technol. **10**(1), 1–309 (2017)
12. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Mach. Learn. **46**(1–3), 389–422 (2002)
13. Honnibal, M., Montani, I.: spaCy 2: natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing (2017, to appear)
14. Milani, A., Santucci, V.: Asynchronous differential evolution. In: Proceedings of 2010 IEEE Congress on Evolutionary Computation (CEC 2010), pp. 1–7 (2010). https://doi.org/10.1109/CEC.2010.5586107
15. Milani, A., Spina, S., Santucci, V., Piersanti, L., Simonetti, M., Biondi, G.: Text classification for Italian proficiency evaluation. In: Misra, S., et al. (eds.) ICCSA 2019. LNCS, vol. 11619, pp. 830–841. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24289-3_61
16. Aprosio, A.P., Moretti, G.: Italy goes to Stanford: a collection of CoreNLP modules for Italian. arxiv e-prints, September 2016
17. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
18. Santucci, V., Spina, S., Milani, A., Biondi, G., Di Bari, G.: Detecting hate speech for Italian language in social media. In: EVALITA 2018, co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), vol. 2263 (2018)
19. Schmidt, A., Wiegand, M.: A survey on hate speech detection using natural language processing. In: Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, pp. 1–10 (2017)
20. Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, Cambridge (2014)
21. Straka, M., Straková, J.: Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with udpipe. In: Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pp. 88–99 (2017)
22. Van der Maaten, L., Weinberger, K.: Stochastic triplet embedding. In: 2012 IEEE International Workshop on Machine Learning for Signal Processing, pp. 1–6, September 2012. https://doi.org/10.1109/MLSP.2012.6349720
23. Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing. IEEE Comput. Intell. Mag. **13**(3), 55–75 (2018)