# Training Readability Comparators for Academic Texts at Different Levels

José Medardo Tapia-Téllez[1], Aurelio López-López[1], Jesús Miguel García-Gorrostieta[2]

[1] Instituto Nacional de Astrofísica Óptica y Electrónica, Puebla, Mexico

[2] Universidad de la Sierra, Sonora, Mexico

{tapiatellez@gmail.com, allopez}@inaoep.mx
jgarcia@unisierra.edu.mx

**Abstract.** Developing machine learning tools to aid students in the process of writing a thesis document is of great interest to students, universities, supervisors and evaluation committees. This article presents the construction and evaluation of readability comparators based in Spanish-written thesis documents of four different academic levels: Advanced College Level Technician (ACT), Undergraduate, Master and Doctoral. Specifically, we provide comparators that can evaluate, between two thesis texts which one is more readable than the other; the thesis sections we focus are: Problem Statement, Results and Justification. The successful completion of these different comparators, as shown in our results, opens the possibility for building a web-based API that analyzes an input thesis draft section and determines whether corresponds to its academic level or requires further improvement.

## 1 Introduction

The quest for aiding students in the production of a thesis document is a longtime problem that affects universities, thesis supervisors, and committees. According to [1], students in general demand flexible forms and structures of research that can explode the extensive use of new technologies. Our aim is to provide the basis to develop these new technologies, specifically for writers in Spanish.

Computational-linguistic technologies such as word correctors or grammar checkers have aided students in the production of a thesis document. We now face a Machine Learning revolution, where tools such as Grammarly[3], advertised as the world's most accurate online grammar checker, assists students and public in general in writing documents of any kind. Such tools are very helpful but have some limitations: first, they were developed for English, and secondly, they are specifically focused on grammar. We need to develop, first these kind of tools for

[3] https://www.grammarly.com

Spanish, and then, we must include deeper linguistic analysis in them. In this work, we report the first steps to reach a readability evaluator of thesis.

According to [2], text readability is the sum of all elements in textual material that affect a reader's understanding, reading speed, and level of interest in the material. Our aim in this work is to build and contrast machine learning-based comparators that are capable to judge between a couple of thesis document sections, indicating which one is more readable than the other. We tackle this task through the use of thesis documents of four different academic levels and we build the comparators based on a previously proposed methodology [9].

The document is organized as follows. Section 2 details related work to our research. Section 3 presents the data-set along with its statistics. Section 4 describes the methodology for the comparators and the experiments. Results and analysis appear in Section 5, and we conclude in Section 6 with future directions.

## 2 Related Work

To obtain an extensive background on how readability of texts is assessed automatically we found [3]. They review the state-of-the-art algorithms in automatic modeling and predicting the reading difficulty of texts, and also list new challenges and opportunities in the area. In general, studies on this area fall within regression and classification, however, one can find research that treats this task as a pairwise problem. This paper allowed to point out the not so explored idea of utilizing comparators as a machine learning tool to evaluate text readability on documents written in Spanish.

The work of [9] successfully builds a comparator through machine learning that can judge text readability between two texts. They then utilize this comparator to sort a set of texts and present an application which retrieves texts with readability similar to that of a given input text. Here, we employ this comparator construction scheme but we implement it for Spanish-written theses and focus on specific sections.

To build our comparator, a document representation suited for our type of documents is needed. In [6], they combine lexical, syntactic, and discourse features to produce a highly predictive model of human reader's judgments of text readability. This is a work that treats readability prediction as a pairwise preference learning problem, thus estimating the relative difficulty of pairs of documents instead of assigning a specific level. Through this work, we could obtain state-of-the-art feature representation for our comparator.

Finally, since our interest is to use the comparators as tools to help students in the process of thesis writing, papers as [7] are pertinent. They evaluate the quality of eBooks through text analytics and identify parts that need improvement. In [4], they theorize about the generation of an automatic validation for a text that contains information pertaining to a medical procedure, and give future approaches on why these tools are viable and important subject of research.

## 3 Document Collection

For our research, we use the document collection of [5]. The data set consists of theses and proposals of different academic levels, such as: Advance College-level Technician[4] (ACT), Undergraduate, Master, and Doctoral. The following sections were extracted from each document: Problem Statement, Justification, and Results. Table 1 includes the number of theses of each level in the collection (All), the selected theses (i.e. documents with more than one paragraph), and the number of sections of interest. Word/token statistics for the three sections in the document collection is given in Table 2, that includes average, larger text (Max) and shorter text (Min).

**Table 1.** Collection data by academic level

| | All | Selected | Problem Stmt | Justification | Results |
|---|---|---|---|---|---|
| **ACT** | 227 | 202 | 80 | 104 | 102 |
| **Undergraduate (UG)** | 150 | 136 | 28 | 21 | 77 |
| **Master** | 269 | 254 | 92 | 81 | 179 |
| **Doctoral** | 66 | 64 | 21 | 13 | 43 |

**Table 2.** Statistics of tokens per section and academic level.

| | Problem Stmt | | | Justification | | | Results | | |
|---|---|---|---|---|---|---|---|---|---|
| | Avg. | max. | min. | Avg. | max. | min. | Avg. | max. | min. |
| **ACT** | 409.33 | 1708 | 75 | 385.07 | 1123 | 110 | 493.55 | 1883 | 138 |
| **UG** | 477.60 | 1253 | 114 | 342.33 | 820 | 109 | 508.92 | 2386 | 121 |
| **Master** | 399.39 | 1063 | 121 | 407.81 | 1612 | 113 | 656.65 | 5184 | 137 |
| **Doctoral** | 754.85 | 1643 | 310 | 433.84 | 802 | 247 | 840.27 | 3765 | 59 |

## 4 Methodology

### 4.1 Comparator Construction

Each comparator is trained through machine learning and can judge, given two texts, which one is more readable than the other. To build the comparator, we have to first determine a representation for documents and then train the model.

**Document Representation** Our dataset consists of theses of different academic levels, each thesis is divided into sections and we extract sections of interest from each document. To get a vector to feed in the machine learning model, we must have two texts from different theses, each of the same kind of section but different academic levels. Let call these texts $a, b \in S$, with $S$ the

[4] A two year program offered in some countries

set of texts selected from all the thesis documents. To build the feature vectors $V_a$ and $V_b$, we extract local and global characteristics from $a$ and $b$, as shown in the lower part of Figure 1, where each $V_a$ and $V_b$ has its local and global features. By local characteristics, we mean the frequency of each word divided by the frequency of the number of words in the text, i.e. the relative frequency. By global features, we refer to the log frequency of the 5000 most common words in Spanish[5].

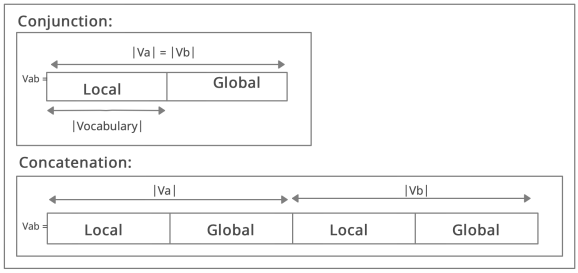


**Fig. 1.** Feature representation for text documents

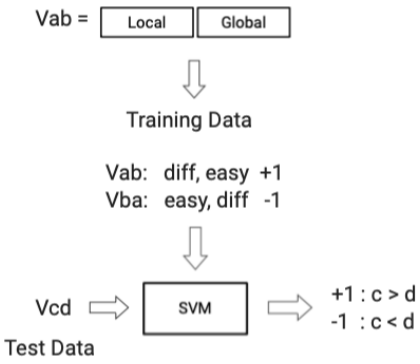


**Fig. 2.** Training and testing diagram for comparator

Now, a single vector from vectors $V_a$ and $V_b$ is obtained by operate them. In Figure 1, we can observe that the conjunction operation of $V_a$ with $V_b$ produces a $V_{ab}$ vector of the same size, since this considers two possible operations: vector difference and vector division. As the lower part of Figure 1, the concatenation operation corresponds to placing together vector $V_a$ and $V_b$, thus resulting in a $V_{ab}$ vector with twice the size of $V_a$ or $V_b$. These two types (Conjunction and Concatenation) of $V_{ab}$ vectors are used to train the different comparator models.

[5] [8] is a dataset created by the Royal Academy of (Spanish) Language, that provides statistics of the most common words.

**Table 3.** Accuracy percentage for comparators trained with vector operated by difference on three different machine learning models across all sections.

| Comparator | Problem Stmt | | | Justification | | | Results | | |
|---|---|---|---|---|---|---|---|---|---|
| | SVM | KNN | Perceptron | SVM | KNN | Perceptron | SVM | KNN | Perceptron |
| **ACT-Undergraduate** | 36.65 | 41.59 | 37.39 | 64.70 | 53.61 | 63.87 | 46.38 | 56.26 | 42.41 |
| **ACT-Master** | 54.41 | 55.55 | 53.18 | 61.99 | 64.93 | 56.97 | 64.14 | 62.17 | 69.43 |
| **ACT-Doctoral** | 81.09 | 55.46 | 82.14 | 83.58 | 67.69 | 73.33 | 82.14 | 65.87 | 81.15 |
| **Undergraduate-Master** | 57.53 | 47.42 | 47.81 | 70.84 | 50.34 | 65.31 | 77.67 | **74.96** | 81.73 |
| **Undergraduate-Doctoral** | **90.58** | **77.64** | **87.64** | **90.58** | **77.64** | **87.64** | **86.11** | 62.64 | **85.64** |
| **Master-Doctoral** | 77.64 | 67.64 | 64.11 | 77.64 | 67.64 | 64.11 | 74.35 | 53.20 | 75.40 |
| **Average** | 66.31 | 57.55 | 62.04 | 74.88 | 63.64 | 68.53 | 71.79 | 62.51 | 72.62 |

**Training the comparator.** $V_{ab}$ vectors come from a set of sections of two different academic levels, let call them $A$ and $B$, with $a \in A$, $b \in B$, $|A| = n$ and $|B| = m$. Each vector of a document in $A$ is operated with each of the vectors of documents in $B$ and classified as $+1$ if $A$ is of a higher academic level than $B$ and $-1$ otherwise, leading to $n \times m$ vectors $V_{ab}$.

We also perform this procedure the other way around, i.e. with $B$ first, and thus obtaining $m \times n$ vectors $V_{ba}$. This is done since $V_{ab}$ vectors are not the same as $V_{ba}$ vectors and their classification differs. So, we end up with a matrix of size $2 \times n \times m$ for training.

This matrix is used to train the selected machine learning models for our comparator, as illustrated in Figure 2, where vectors of the form $V_{ab}$ and $V_{ba}$ are fed into a Support Vector Machine (SVM). After training, Figure 2 also depicts a $V_{cd}$ vector input to the machine, that was created from two texts: $c$ and $d$. These texts were vectorized and operated, leading to the $V_{cd}$ vector which can be fed to the trained SVM to get an evaluation of their relative readability.

### 4.2 Experiments

Given that texts are of four different academic levels, and each of the comparators is built from two different academic levels, we can construct six: ACT-Undergraduate, ACT-Master, ACT-Doctoral, Undergraduate-Master, Undergrad-Doctoral and Master-Doctoral. Likewise, for each thesis document, we extracted three sections of interest: Problem Statement, Justification and Results. Finally, in order to operate the vectors we have three vector operations: division, difference, and concatenation.

We set experiments where for each vector operation, we trained the six possible comparators for each section with three different machine learning models: SVM, K-Nearest Neighbors (K-NN), and Perceptron.

## 5 Results and Analysis

### 5.1 Comparators Trained with Vectors Operated as Difference

In Table 3, we can observe that the comparator with the worst efficacy is ACT-Undergraduate across all sections.

The best result corresponds to Undergraduate-Doctoral comparator followed by ACT-Doctoral. As to average, SVM produces the best averages across Problem Statement and Justification, and Perceptron slightly better in Results

**Table 4.** Accuracy percentage for comparators trained with vector operated as division, with three different machine learning models across all sections.

| Comparator | Problem Stmt | | | Justification | | | Results | | |
|---|---|---|---|---|---|---|---|---|---|
| | SVM | KNN | Perceptron | SVM | KNN | Perceptron | SVM | KNN | Perceptron |
| **ACT-Undergraduate** | 48.73 | 53.15 | 42.33 | 54.97 | 49.69 | 50.00 | 41.97 | 56.61 | 43.38 |
| **ACT-Master** | 58.61 | 51.06 | 61.31 | 59.35 | 52.79 | 58.74 | 65.45 | 64.74 | 66.59 |
| **ACT-Doctoral** | 65.33 | 56.51 | 50.00 | 71.53 | **60.00** | 50.00 | 82.14 | 64.88 | 78.67 |
| **Undergraduate-Master** | 57.93 | 45.73 | 55.65 | 59.25 | 49.13 | **61.24** | 72.47 | **71.40** | 71.54 |
| **Undergraduate-Doctoral** | **65.81** | 58.16 | **66.83** | **73.52** | 50.58 | 50.00 | **82.25** | 68.05 | **83.48** |
| **Master-Doctoral** | 62.50 | **59.72** | 50.00 | 57.35 | 44.41 | 50.00 | 69.55 | 53.20 | 73.39 |
| **Average** | 59.81 | 54.05 | 54.35 | 62.66 | 51.10 | 53.33 | 68.97 | 63.14 | 69.50 |

**Table 5.** Accuracy percentage for comparators trained with vectors operated as concatenation, with three different machine learning models across all sections.

| Comparator | Problem Stmt | | | Justification | | | Results | | |
|---|---|---|---|---|---|---|---|---|---|
| | SVM | KNN | Perceptron | SVM | KNN | Perceptron | SVM | KNN | Perceptron |
| **ACT-Undergraduate** | 35.08 | 44.53 | 41.38 | 64.55 | 48.26 | 65.38 | 45.41 | 61.19 | 44.22 |
| **ACT-Master** | 52.77 | 56.29 | 61.35 | 61.76 | 53.77 | 56.90 | 65.65 | 66.84 | **62.43** |
| **ACT-Doctoral** | 79.83 | **57.14** | **86.97** | 83.58 | 68.71 | 87.69 | 80.35 | **68.25** | 50.00 |
| **Undergraduate-Master** | 57.34 | 52.08 | 68.45 | 71.19 | 47.75 | 72.31 | **80.74** | 67.29 | 60.23 |
| **Undergraduate-Doctoral** | **74.48** | 40.81 | 64.79 | **88.23** | **75.29** | **80.58** | 80.24 | 66.35 | 50.00 |
| **Master-Doctoral** | 69.04 | 50.79 | 79.16 | 78.82 | 61.17 | 64.41 | 67.78 | 56.25 | 50.00 |
| **Average** | 61.42 | 50.27 | 67.01 | 74.68 | 59.15 | 71.21 | 70.02 | 64.36 | 52.81 |

### 5.2 Comparators Trained with Vectors Operated as Division

Table 4 shows that the comparator that performed worst across all sections is again ACT-Undergraduate. The comparator with the best results is ACT-Doctoral followed by Undergraduate-Doctoral. As for average of models, SVM has the best results in Problem Statement and Justification sections. In Results, Perceptron again slightly outperforms SVM.

### 5.3 Comparators Trained with Vectors Operated as Concatenation

We can notice in Table 5 that the comparator with the worst results across all sections is ACT-Undegraduate. On the contrary, the comparators with the best results are Undergraduate-Doctoral followed by ACT-Doctoral. As for the averages of machine learning models, Perceptron achieves the best results for Problem Statement section and SVM for the Justification and Results sections.

### 5.4 Result Analysis and Discussion

General behaviors observed across Tables 3, 4 and 5 are: The comparator with the best results is Undergraduate-Doctoral followed by ACT-Doctoral; and the comparator with the worst results is ACT-Undergraduate. As for operators, we

can observe that difference operator obtains on average the best results; followed by concatenation and lastly division. Based on machine learning models, SVM obtains the best results in two out of three sections, in the three experiments.

Perceptron performs well three times (twice in Results and one in Problem Statement). K-NN did not compete. So, SVM showed a more consistent performance. As final remarks, further experimentation with concatenation operation is not viable since takes too much time, thus we conclude that the difference operator is our ideal operator. Based on comparators, a nice observed feature is that comparators trained with more distant academic levels obtain higher classification accuracy values, e.g. ACT-Undergraduate is lower on accuracy results than ACT-Doctoral. This one being an important observation, since it validates our results and provides insight into the learning capabilities of our machine learning models.

## 6 Conclusion

The present work successfully built and evaluated machine learning based readability comparators trained with Spanish-written thesis documents of four different academic levels. These comparators can decide between two different thesis sections, which one is more readable. In general, results across all sections showed that the best comparator was Undergraduate-Doctoral, the most stable model to train them was SVM, and representation with difference operator is both efficient and lightweight. Based on this information, we can conclude that these types of comparators can definitely serve as a base tool, not only to compare readability between sections of documents, but also to evaluate the academic level of the document.

Comparators have limitations, and though we present comparators with good accuracy level, we have to design a process to employ them to assess new documents in progress (drafts). So as future work, we plan the creation of evaluators, where we will take a representative document for each academic level, and use it to evaluate the level of a draft. We will also explore the possibility of incorporating a pairwise ranking model as that of [10], that brings deep learning in our research. Finally, we are interested in deploying our machine learning-based comparator in a web-based API that, given an input document, estimates its readability level.

**Acknowledgments.** First author was supported by Conacyt through scholarship number 1009285. Second and third authors were partially supported by SNI, México.

## References

1. de Anglat, H.D.: Las funciones del tutor de la tesis doctoral en educación. Revista Mexicana de Investigación Educativa 16(50), 935–959 (2011)
2. Chall, J.S., Dale, E.: Readability revisited: The new Dale-Chall readability formula. Brookline Books (1995)

3. Collins-Thompson, K.: Computational assessment of text readability: A survey of current and future research. ITL-International Journal of Applied Linguistics 165(2), 97–135 (2014)
4. Glaser, I., Bonczek, G., Landthaler, J., Matthes, F.: Towards computer-aided analysis of readability and comprehensibility of patient information in the context of clinical research projects. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. pp. 260–261 (2019)
5. González-López, S., López-López, A.: Colección de tesis y propuesta de investigación en tics: un recurso para su análisis y estudio. In: XIII Congreso Nacional de Investigación Educativa. pp. 1–15 (2015)
6. Pitler, E., Nenkova, A.: Revisiting readability: A unified framework for predicting text quality. In: Proceedings of the 2008 conference on empirical methods in natural language processing. pp. 186–195 (2008)
7. Relan, M., Khurana, S., Singh, V.K.: Qualitative evaluation and improvement suggestions for ebooks using text analytics algorithms. In: Proceedings of Second International Conference on Eco-friendly Computing and Communication Systems, Solan, India (2013)
8. Sánchez, M.S., Cintas, C.D.: El banco de datos de la rae: Crea y corde. Per Abbat: boletín filológico de actualización académica y didáctica (2), 137–148 (2007)
9. Tanaka-Ishii, K., Tezuka, S., Terada, H.: Sorting texts by readability. Computational linguistics 36(2), 203–227 (2010)
10. Wang, L., Shen, X., de Melo, G., Weikum, G.: Cross-domain learning for classifying propaganda in online contents. arXiv preprint arXiv:2011.06844 (2020)