How to simplify human-machine interaction

A text complexity calculator and a smart spelling corrector

Licia Sbattella
Politecnico di Milano - DEIB
Milano, Italy
licia.sbattella@polimi.it

Roberto Tedesco
Politecnico di Milano - MultiChancePoliTeam
Milano, Italy
roberto.tedesco@polimi.it

ABSTRACT

The most basic human-machine interactions are about reading text and typing on a keyboard. Users face two fundamental issues: on one hand, complex text can prevent users from accessing information, on the other hand the average user tends to make several typos. And these issues are even harder for fragile persons. We propose two tools that could ameliorate that situation: a text complexity evaluator and a smart spelling corrector.

CCS CONCEPTS

• **Human-centered computing** → *Accessibility systems and tools*;

KEYWORDS

Text complexity, spelling correction

ACM Reference Format:
Licia Sbattella and Roberto Tedesco. 2018. How to simplify human-machine interaction: A text complexity calculator and a smart spelling corrector. In *International Conference on Smart Objects and Technologies for Social Good (Goodtechs '18), November 28–30, 2018, Bologna, Italy*. ACM, New York, NY, USA, Article 4, 2 pages. https://doi.org/10.1145/3284869.3284923

1 INTRODUCTION

Although Graphical UIs and Touch UIs are nowadays the standard HCI adopted by almost all ICT devices, the most basic human-machine interactions are still about reading text displayed on a screen and typing on a (eventually, virtual) keyboard.

Developers and authors should provide readable text, to avoid precluding fragile persons (for example, people with dyslexia, cognitive problems, low vision and, of course, the elderly) from accessing information. We propose a complexity evaluator that developers and authors could use for writing good pieces of text.

On the other hand, typing on a keyboard in an effective way is not an easy task. For persons with motor disability, cognitive problems, and the elderly, using effectively a keyboard is even more difficult. Solving this problem is about reducing the effort required for fixing typos. We propose an advanced spelling corrector that is able to correct a whole sentence at at time, dramatically reducing the user-machine interactions needed by conventional tools.

Goodtechs '18, November 28–30, 2018, Bologna, Italy

ACM ISBN 978-1-4503-6581-9/18/11…$15.00
https://doi.org/10.1145/3284869.3284923

2 ON TEXT COMPLEXITY: SPARTA2

Classical approaches, still in use today, rely on simple formulas, based on word or sentence length (e.g., the Flesch-Kincaid index.)

SPARTA2 (*Sistema per la Produzione Assistita e Revisione di Testi ad Alta Accessibilità*), in contrast, makes a distinction between *readability* and *understandability* as, in our opinion, these concepts capture different aspects of the complexity of the text: a text could be highly readable, since the syntax is extremely simple, but extremely hard to understand because of the lexicon used. In our approach, readability gives an evaluation about the structure of sentences, while understandability captures the lexical aspects. Details about the algorithm can be found in [3].

Understandability. This complexity is based on the De Mauro's Italian dictionary [1], which contains the 4700 most used lemmas of the Italian language. The vocabulary is divided into three sections: *(1) basic words*, *(2) highly used words*, and *(3) less used words*. Tokens non appearing into the De Mauro' dictionary are classified as highly complex. Then, given a sentence s, its understandability is

$$\text{Understandability}(s) = \frac{\alpha \cdot N_1 + \beta \cdot N_2 + \gamma \cdot N_3}{W} \quad \in [0,1] \qquad (1)$$

where N_1, N_2, and N_3 are the number of tokens of s belonging to the vocabularies (1), (2), and (3) , while W is the number of tokens of s. Parameters α, β, and γ are set to 1, 0.75, and 0.5. Notice that the formula does not consider the N_x tokens not belonging to the dictionary, because we consider such tokens as having a parameter equals to zero; thus, $W = N_1 + N_2 + N_3 + N_x$. Values near 1 indicate high understandability.

Readability. This complexity is composed by three sub-indexes.

The Gulpease Index [2] is a readability formula for the Italian language, similar to the Flesch-Kincaid index

$$G = \min(89 - 10 \cdot \frac{N_C}{N_W} + 300 \cdot \frac{N_S}{N_W}, 100) \quad \in [0,100] \qquad (2)$$

where N_C is the number of character in the text, N_W the number of tokens, and N_S the number of sentences.

The Chunk Index relates the number of chunks in a text to its readability (we argue that a sentence with a few chunks should be more readable)

$$K = \begin{cases} \frac{1}{\frac{N_K}{N_S} - 1}; & N_K > 1 \\ 1; & N_K = 1 \end{cases} \quad \in [0,1] \qquad (3)$$

where N_K is the number of chunks in the text.

However, using the number of chunks does not consider the fact that different chunk types (Nominal Phrase, Verbal Phrase, etc.) could have different readability. Thus, we added the Chunk Type Index, based on the distribution of chunk types T in the text

$$K_T = \begin{cases} 5.44 \cdot \frac{\sum_i^T w_i \cdot N_i}{N_K}; & N_K > 1 \\ \frac{\sum_i^T w_i \cdot N_i}{0.2468}; & N_K = 1 \end{cases} \in [0, 1] \quad (4)$$

where N_i is the number of chunks of type i in the text, and w_i the corresponding weight. The simplest text, composed of a Nominal Phrase and a Verbal Phrase, gets the maximum value.

The weights w_i were calculated analyzing the statistical distribution of chunk types we found in the AltaFrequenza on-line magazine, which publishes carefully written, simplified news. The readability is then computed as

$$\text{Readability}(s) = a \cdot \frac{G}{100} + b \cdot K + c \cdot K_T \in [0, 1] \quad (5)$$

where $a = b = c = 1/3$. Values near 1 mean high readability.

SPARTA2 has been implemented as a Word plug-in, and tested; statistical significant data are still to be collected, however.

3 ON SPELLING CORRECTION: IESO

Lowering the effort needed to input correct text is crucial for providing a good user experience. Classical approaches correct one word at a time generating, for each error, a list of possible corrections.

IESO (Intelligent Emendation of Spelling Oversights), in contrast, corrects a whole sentence at a time. It is composed by an encoder, which consists of two LSTM (Long-Short Term Memory) layers, an attention layer, and a decoder made by two LSTM layers.

For the training phase the samples are composed of paired sentences: the sentence with errors and its corrected version. In the decoding stage, IESO receives a misspelled sentences and returns a possible suggestion for the corrected version.

For training IESO we developed an ad hoc corpus, starting from a huge collection of correct sentences (594 317 sentences, containing 13 848 568 tokens and 254 937 word types) and generating a "wrong version" adding three error typologies: typo errors, due to the keyboard layout; grammatical errors, like incorrect verb tenses or word forms; and complex errors (word splitting/merging, character swapping, etc.) Adding such errors generated new word types (the "wrong version" dataset contained 1 083 856 word types).

For the testing phase we divided our corpus into 5 sets to evaluate IESO with different sentence lengths: Set 1 (words with length shorter or equal to 5 characters), Set 2 (6 to 10), Set 3 (11 to 15), Set 4 (16 to 20), and Set 5 (21 to 25).

In Figure 1, a zero edit distance corresponds to the accuracy; as expected, it decreases with the length of the sentence. In general, as expected, the distance increases with the sentence length.

In Table 1, IESO corrects all the errors with the exception of the word *stagnze*, which should be corrected as *stanze*. The correction

Table 1: An example (typos are underlined)

Typed	*le varie stagnze dell'edififcio sono conservafte discretamen teed è possibile accedervi per compeendere*
True	*le varie* ***stanze*** *dell'****edificio*** *sono* ***conservate discretamente ed*** *è possibile accedervi per* ***comprendere***
IESO	*le varie* FRAZIONI *dell'****edificio*** *sono* ***conservate discretamente ed*** *è possibile accedervi per* ***comprendere***

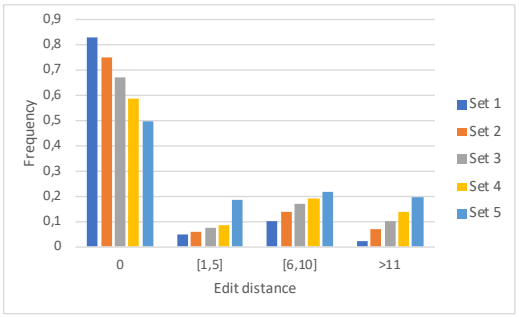

Figure 1: Edit distance; ground-truth vs corrections.

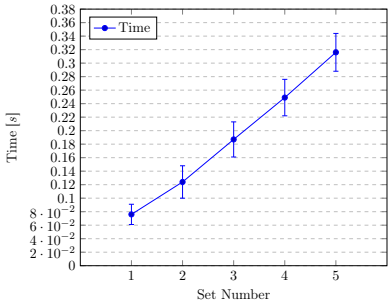

Figure 2: Correction time, for each set (error bars: std. dev.)

is *frazioni*, which is wrong but brings a similar meaning: *stanze* (rooms) are parts of a house and *frazioni* (fractions) are parts of something. This example shows the context sensitivity of IESO and a side effect (misleading, in this example) of word embeddings.

Finally, Figure 2 shows the performances of IESO (TensorFlow on Intel(R) Xeon(R) E5-260 i5 CPU, 4 core, 2.5 GHz, 32GB RAM, Nvidia Titan X GPU).

4 CONCLUSION

In this paper we have showed a text complexity evaluator and an advanced spelling corrector we developed for assisting developers/authors and final users. Our tools, although promising, are far from being perfect. In particular, the complexity evaluator needs further research for choosing the best values for its parameters, and more experiments are needed for developing a reference scale for the indexes it calculates. The spelling corrector needs an effective way for adapting IESO to the user's most frequent mistakes.

ACKNOWLEDGMENTS

The authors would like to thank A. Colombo and M. Pagliari, who worked on SPARTA2 and IESO for their graduate theses. This work is part of the LYV Project (PoliSocial Award 2015/2016.)

REFERENCES

[1] T. De Mauro and G.G. Moroni. 1996. *DIB - Dizionario di base della lingua italiana.* Paravia.
[2] P. Lucisano and M.E. Piemontese. 1988. GULPEASE: una formula per la predizione della difficoltà dei testi in lingua italiana. *Scuola e città* 3, 31 (1988).
[3] L. Sbattella and R. Tedesco. 2012. Calculating text complexity during the authoring phase. Proceedings of the W3C RDWG Easy-to-Read on the Web Online Symposium. http://www.w3.org/WAI/RD/2012/easy-to-read/paper3/