Matching Readers with Text through Readability Assessment

Mohamed Sayed Mohamed Magdy

Shehab Khaled

{mohamed.sayed, mohamedmagdi14, shehabeldeen555}

@stud.fci-cu.edu.eg

June 25, 2019

Contents

List of Figures

List of Tables

Glossary

ARA Automatic Readability Assessment. 6

ATS Automatic Text Simplification. 24

CEFR The Common European Framework of Reference for Languages. 17

CNN Convolution Neural Network. 2, 20, 21, 23

FF Feed Forward. 2, 17

LSTM Long Short Term Memory. 2, 19, 23

NLTK Natural Language Processing Toolkit. 9

POS Part of Speech. 2, 4, 17, 20, 23

seq2seq Sequence to sequence. 24

Chapter 1

Introduction

1.1 Problem definition

At first readability is the ease with which a reader can understand a written text. In natural language, the readability of text depends on its content (the complexity of its vocabulary and syntax) and its presentation (such as typographic aspects like font size, line height, and line length). Higher readability eases reading effort and speed for any reader, but it is especially important for those who do not have high reading comprehension. In readers with average or poor reading comprehension, raising the readability level of a text from mediocre to good can make the difference between success and failure of its communication goals.

Automatic Readability Assessment (ARA), the task of assessing the reading difficulty of a text, is a well-known problem in computational linguistics. Assigning an appropriate metric that could match the reader with the most suitable material remains a hurdle. As most of the readability assessment tools used today are formulas that rely on the number of words, the number of sentences and their average

lengths which aren't accurate measurements. As a short word can be difficult if it is not used very often by most people. The frequency with which words are in normal use affects the readability of text. And by just counting the number of words per sentence we don't define the structure of the sentence.

In this background, we tried a different approach to assess the readability of the text by using natural language processing and deep learning as we found a corpus aligned at text and sentence level, across three reading levels (beginner, intermediate, advanced). the corpus contains articles from The Guardian newspaper and rewritten by teachers to suit the three levels. So we built a neural network using this corpus to achieve more accurate assessments.

For the ease of using the tool, we developed a browser extension to make the user able to select an article in a newspaper website and get the reading difficulty of that article just by clicking on the extension icon. And the user can type an article to get it's reading difficulty.

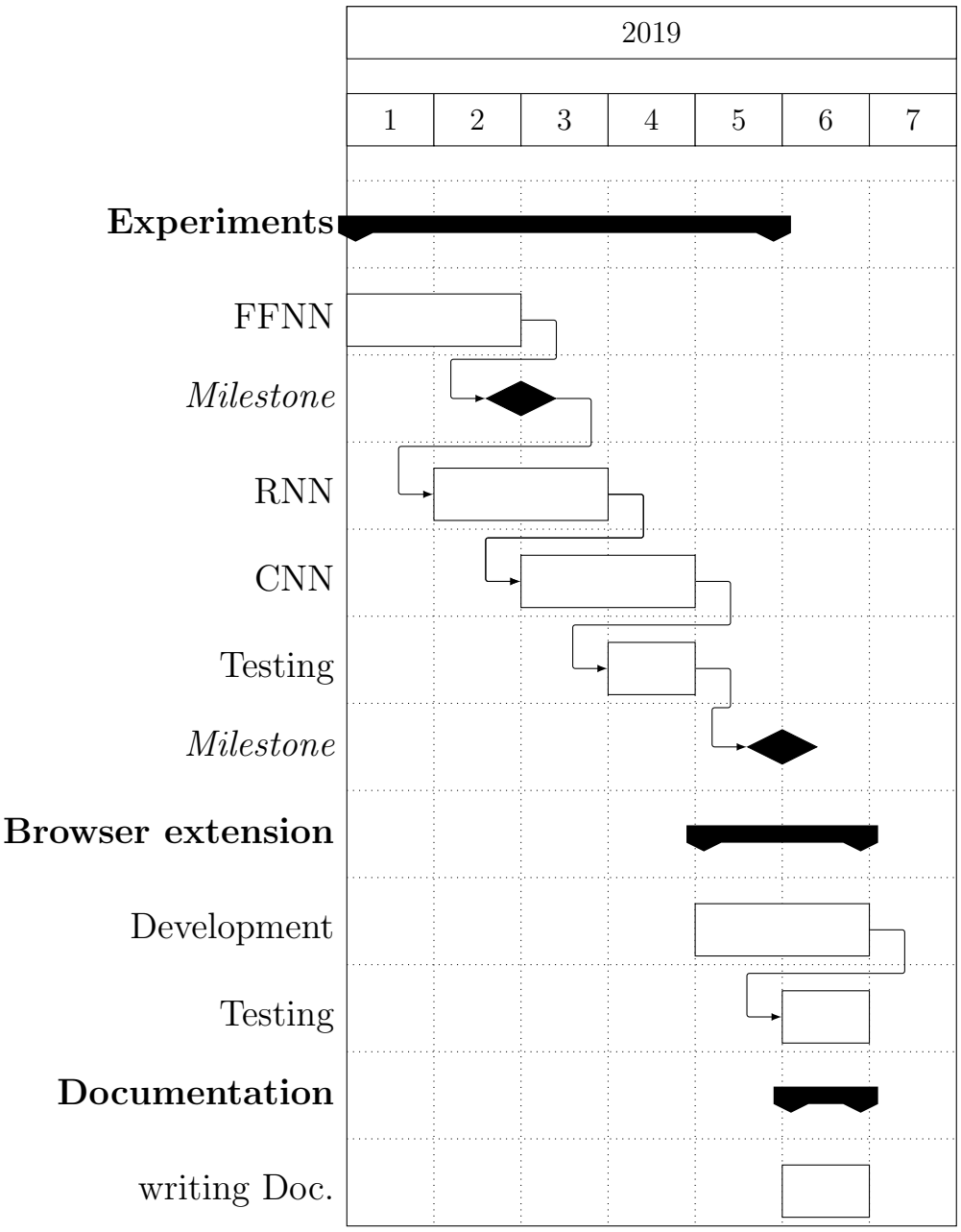
1.2 Motivation

Facilitates the process of automatic readability assessment and achieves this with high accuracy, that helps English learners to determine the reading difficulty of articles in newspaper websites.

1.3 Project Objective

Building an automatic readability assessment tool using natural language processing and deep learning.

1.4 Gantt chart of project time plan



1.5 The used tools in the project

In this section we states the tools that helped us to complete our project.

1. **Python**[4]: Programming language we used in developing learning models and developing the back end of the browser extension.

2. **Keras**[5]: Keras is an open-source neural-network library written in Python. Designed to enable fast experimentation with deep neural networks.

3. **NLTK**[6]: Open-source framework used in NLP tasks. We used it to assign each word to its part of speech tag.

4. **Scikit-learn**[7]: Machine learning library. We used it for K-fold cross validation.

5. **Github**: Version control system that enables collaboration online.

6. **LaTeX**[8]: A document preparation system that this document is built on.

7. **Trello**: A flexible way to organize plans and tasks online.

1.6 Report Organization

In this document we are going to explore the readability assessment task. We state the tools we used, references we conducted and most importantly the experiments that we have done using different techniques and the accuracy we achieved through this experiments.

Chapter 2

Related Work

there are various readability assessment formulas which depend on a small number of features from the given paragraph so these formulas have limitations and the accuracy of their assessment isn't high examples of these formulas:

1. Gunning fog index[9]

    - the Gunning fog index is a readability test for English writing.
    - The index estimates the years of formal education a person needs to understand the text on the first reading.
    - The Gunning fog index is calculated with the following algorithm:

        i Select a passage (such as one or more full paragraphs) of around 100 words. Do not omit any sentences;

        ii Determine the average sentence length. (Divide the number of words by the number of sentences.);

        iii Count the "complex" words consisting of three or more syllables. Do not include proper nouns, familiar jargon,

or compound words. Do not include common suffixes (such as -es, -ed, or -ing) as a syllable;

iv Add the average sentence length and the percentage of complex words; and

v Multiply the result by 0.4.

- The complete formula is:

$$0.4\left[\left(\frac{words}{sentences}\right) + 100\left(\frac{complex\ words}{words}\right)\right] \tag{2.1}$$

- Limitations
  - While the fog index is a good sign of hard-to-read text, it has limits. Not all complex words are difficult. For example, "interesting" is not generally thought to be a difficult word, although it has four syllables. A short word can be difficult if it is not used very often by most people. The frequency with which words are in normal use affects the readability of text.

2. Flesch–Kincaid readability tests[10]

   - The Flesch–Kincaid readability tests are readability tests designed to indicate how difficult a passage in English is to understand. There are two tests, the Flesch Reading Ease, and the Flesch–Kincaid Grade Level. Although they use the same core measures (word length and sentence length), they have different weighting factors[11].

     i Flesch reading ease
       - In the Flesch reading-ease test, higher scores indicate material that is easier to read; lower numbers mark passages that are more difficult to read. The

formula for the Flesch reading-ease score (FRES) test is

$$206.835 - 1.015\left(\frac{total\ words}{total\ sentences}\right) - 84.6\left(\frac{total\ syllables}{total\ words}\right) \tag{2.2}$$

ii Flesch–Kincaid grade level

- The "Flesch–Kincaid Grade Level Formula" instead presents a score as a U.S. grade level, making it easier for teachers, parents, librarians, and others to judge the readability level of various books and texts. It can also mean the number of years of education generally required to understand this text, relevant when the formula results in a number greater than 10. The grade level is calculated with the following formula:

$$0.39\left(\frac{total\ words}{total\ sentences}\right) + 11.8\left(\frac{total\ syllables}{total\ words}\right) - 15.59 \tag{2.3}$$

Chapter 3

System Analysis

3.1 Functional Requirement

Introduction This is the functional requirements specification for Matching readers with text, contains details of the capabilities and functions that the Matching readers with text must be capable of performing, These requirements will assure that the software will correctly and reliably perform its intended functionality.

Matching Readers with text is a software that take any text as an input and tells the user what difficulty level this text at according to our One Stop English dataset, which provide three levels(Elementary, Intermediate, and Advanced) we'll see it in details in the next Chapter. The output will be one of the three levels telling the user the difficulty level of the text he's entered.

3.2 Non-functional Requirements

- Efficiency

  The output must be ready in less than 1 minute.

- Portability

  The software must be multi platform and doesn't require specific hardware components.

- Availability

  The software must be up as long as there is internet connection.

- Usability

  The software user interface must be easy to interact with. So it could be used by people with different ages. In case of errors and system failures the user must be informed.

Chapter 4

Corpora

Now for training and testing our assessments models we're explored two corpora.

4.1 Common European Framework of References

CEFR is an international standard for describing language ability. It describes language ability on a six-point scale, from A1 for beginners, up to C2 for those who have mastered a language. This makes it easy for anyone involved in language teaching and testing, such as teachers or learners, to see the level of different qualifications. It also means that employers and educational institutions can easily compare our qualifications to other exams in their country. Our dataset is group of 183 articles varies in length. Those articles are labeled through CEFR framework. Dataset is gathered by hand from several websites.

4.2 One Stop English[1]

This corpora is consisting of 567 that are parallel from 189 at three reading levels(Beginner, intermediate and advanced). The articles are originally collected from Guardian newspaper and manually rewritten by experts for teaching at three levels. The corpora demonstrates its usefulness for two applications automatic text simplification and automatic readability assessment and that's the problem we're facing and trying to solve, ARA is the task of assessing a reading difficulty of a text.

4.3 Domain Adaption

Supervised machine learning and statistical methods like the ones used in this paper benefit from the availability of large amounts of training data. However, in many cases it is not easy to obtain enough training data for specific domains or applications. As a result it is not uncommon that researchers train on data from one domain and test on data from a different one. As would be expected, this usually leads to lower levels of performance. The field of domain adaptation is addressing this problem by proposing methods that can perform well even when the training and testing domains are different. Our data is not specified in a domain it consists of articles in newspapers in different domains and different topics, it provides a kind of generalization to assesses the text, also if the input data in specific domain with specific details don't expect an efficient results.

Chapter 5

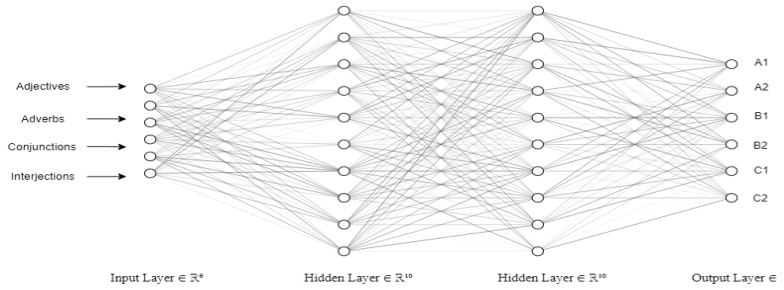
Experiments

In this chapter we are explaining the different experiments we have conducted on our corpora with different techniques.

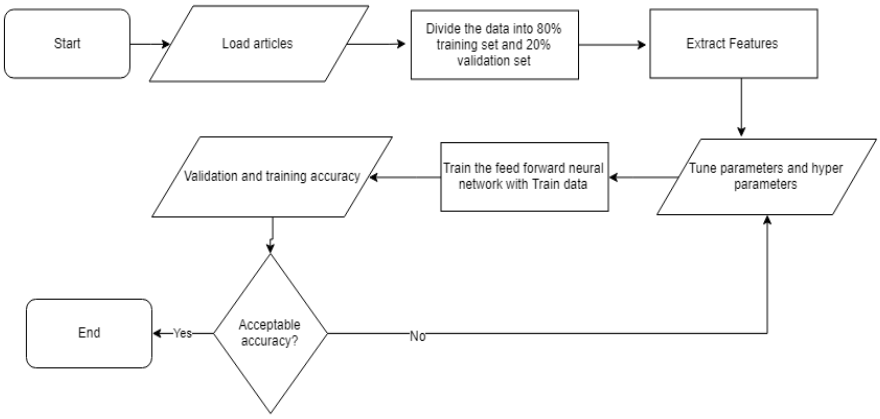
5.1 FF model with POS frequencies

This is our first model we have implemented and discussed in the mid year discussion. We have used the CEFR corpus that we have discussed in the corpora section. In this model we decided to extract 12 features from the text and then embed those features in a classical feed forward neural network. Those features are frequencies of:

Adjectives	Adverbs
Articles	Nouns
Conjunctions	Interjections
Numerals	Past participle verbs
Prepositions	Symbols
Pronouns	Punctuation



(a) Feedforward neural network



(b) Flowchart of FF neural network

Figure 5.1: Feedforward neural network

We have reached a training accuracy of 91.1% and a validation accuracy of 54.05%.Those features were suggested a long with another 40 features in that paper[12]. As we see the validation accuracy is so small compared with training accuracy which reflects what is called over fitting. Also one of the committee's feedback was that the frequencies of the pos do not reflect the readability of the text as extracting frequencies destroys the sentence structure and leave us with bunch of

descriptive statistics about the corpus. We took this feedback into our consideration while making the rest of experiments.

5.2 Bidirectional LSTM[2] model with word embedding

In this model we are trying to overcome the feature extraction problem. Also there are some textual properties such as topic continuity that could not be represented through counting words and sentences.

One of the problems with the first model is the scarcity of the data for each class needed for training. To overcome this problems we have used a new corpus called OneStopEnglish[1] corpus. Our approach this time was a bit different. What is special about LSTM is that it has a memory so their output will depends on the current input and the past input which makes it the most suitable for capturing sentence structure.

1. We first start by splitting the articles into stream of words. This process is called word tokenization.

2. Each words is converted to its Glove word embedding. Glove embedding[13] is a dense representation to their relative meaning of the word. Each word is represented by a list of floating points. Those numbers do reflect the meaning of each word. The word embedding of woman is so similar to the word embedding of girl when calculating the cosine similarity of the two words.

3. Bidirectional LSTM: What is special about the LSTM is that its objective function is based one the current input and the previous one so that the ordering of the sequence of words is maintained.

Also we used the bidirectional version of the network to learn more parameters through mapping the statistical structure of the written language.

Despite the complex structure and the slow training process, this model did achieve 33% and 33% as training and validation accuracy respectively. One of the reasons of this result is that the three versions of each article is are so similar in terms of words. We conclude that we cannot use words to as features in a model that classify text according its style rather than content.

5.3 CNN[3] model with POS sequences

As we said we want now to classify text according to its style rather than its content. To achieve that we want to treat each word as a part of speech. Another problem we faced before is that LSTM network takes a lot of time and computational resources. A good alternative to the LSTM is the CNN. CNN has proved to achieve good accuracy with sequence processing with a moderate usage of computational resources.

1. We first start by splitting the articles into stream of words. This process is called word tokenization.

2. Now each word is converted to its part of speech tag like if it is noun or verb. This process is called POS tagging.

3. We give each part of speech a unique id.

4. CNN: The single dimensional convolution layer can recognize local patterns in the sequence so that it can differentiate between each readability class.

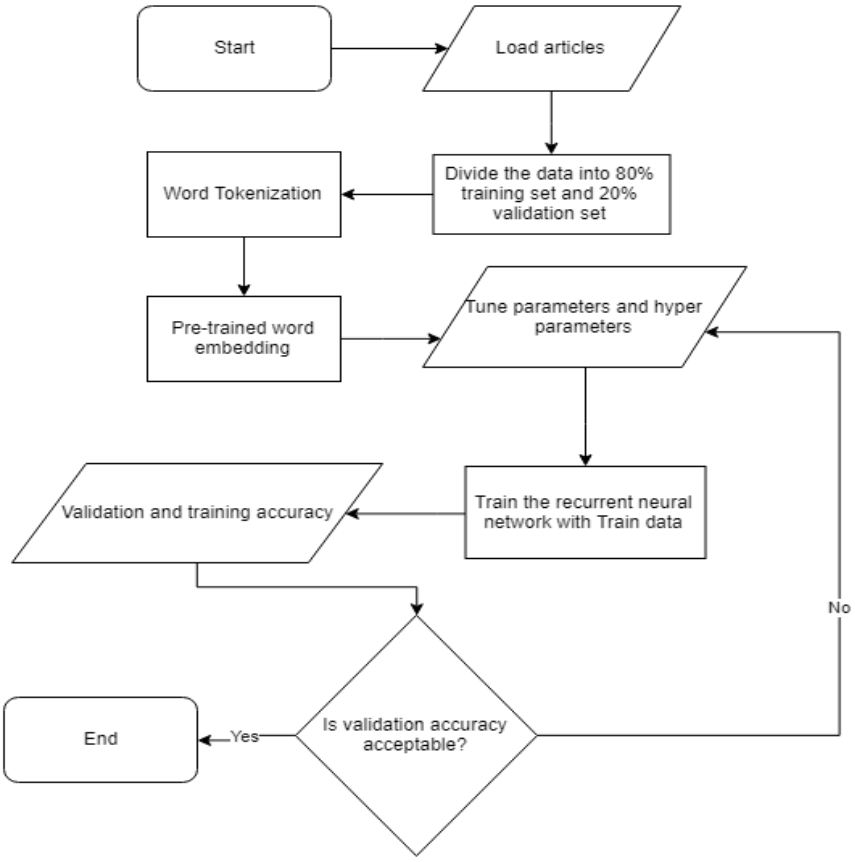


Figure 5.2: Recurrent neural network with LSTM

There are a lot of parameters that are used to tune any neural network model like number of layers, number of neurons and size of moving window of the CNN. There is no way to choose the best parameters that suit your model without trying a lot of them. After a lot of trials we reached a 80% and 76% as training and validation accuracy respectively.

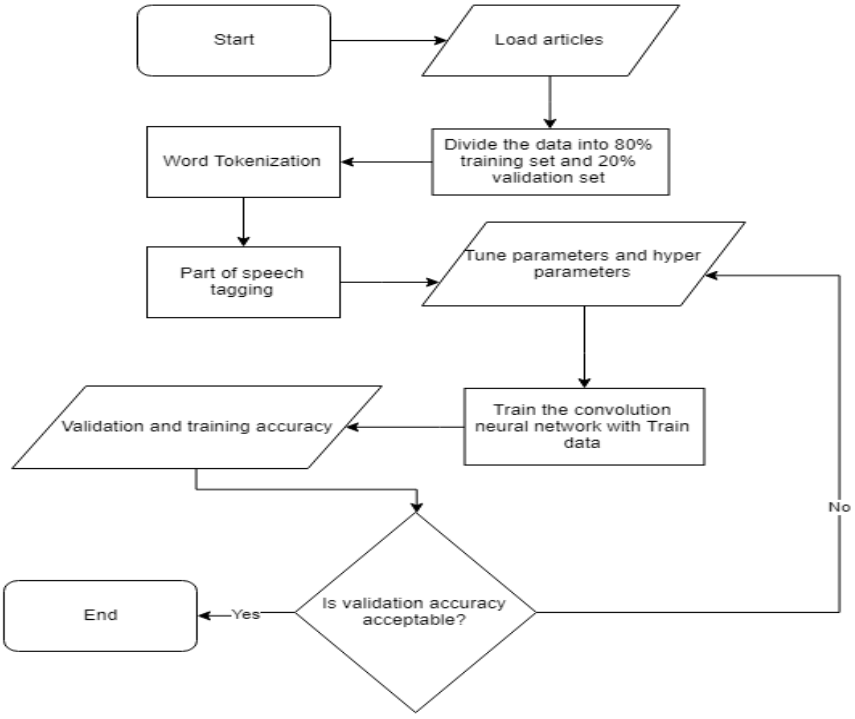


Figure 5.3: Flowchart for Convolution neural network

5.4 Evaluation Metrics

Now in order to evaluate our model we have two types of accuracies.

Training Accuracy	Validation Accuracy
Calculated over training data	Calculated over validation data
80% of the whole data set	20% of the whole data set
Reflects the optimization of the model on the seen data	Reflects the generalization of the model on the unseen data

A huge gap between both accuracies implies the model is over fitting. Now for the final evaluation we use a technique called 10-Fold-cross-validation. What happens is that the model is trained for 10 times. Every time the model is being trained and validated on different portions of the data. At the end the final accuracy is calculated as the mean of the 10 experiments. This approach ensures that the model has got most of the patterns.

5.5 Summary

Model	Accuracy
FF model with POS frequencies	54%
Bidirectional LSTM with word embeddings	33%
CNN model with POS sequences	76%

Chapter 6

Conclusion

In this document we tried to solve readability assessment problem through learning from already classified texts according to its difficulty. We tried different neural network architectures. Although we did not improve the current accuracy that is made by the corpus author[1] but we have been introduced to the field less than a year.

Future Work Also having a parallel corpus opens new opportunities for other readability tasks like ATS. Using Seq2seq model we can build a text simplification model. This process is used in Machine Translation tasks using parallel sentences in source language and target language.

References

[1] Sowmya Vajjala and Ivana Lucic. Onestopenglish corpus: A new corpus for automatic readability assessment and text simplification. 2018.

[2] Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*, 2016.

[3] Cicero Dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, 2014.

[4] Guido Van Rossum and Fred L Drake Jr. *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[5] François Chollet et al. Keras. `https://keras.io`, 2015.

[6] Edward Loper and Steven Bird. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[7] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter

Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[8] Leslie Lamport. *LATEX: a document preparation system: user's guide and reference manual*. Addison-wesley, 1994.

[9] Robert Gunning. The fog index after twenty years. *Journal of Business Communication*, 6(2):3–13, 1969.

[10] Rudolf Flesch. Flesch-kincaid readability test. *Retrieved October*, 26:2007, 2007.

[11] R. Timothy Rush. Assessing readability: Formulas and alternatives. *ERIC*, 19:1984, 1984.

[12] Pedro Curto, Nuno J Mamede, and Jorge Baptista. Automatic text difficulty classifier-assisting the selection of adequate reading materials for european portuguese teaching. In *CSEDU (1)*, pages 36–44, 2015.

[13] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.