Multiattentive Recurrent Neural Network Architecture for Multilingual Readability Assessment

Ion Madrazo Azpiazu and **Maria Soledad Pera**
Department of Computer Science
Boise State University
{ionmadrazo,solepera}@boisestate.edu

Abstract

We present a multiattentive recurrent neural network architecture for automatic multilingual readability assessment. This architecture considers raw words as its main input, but internally captures text structure and informs its word attention process using other syntax- and morphology-related datapoints, known to be of great importance to readability. This is achieved by a multiattentive strategy that allows the neural network to focus on specific parts of a text for predicting its reading level. We conducted an exhaustive evaluation using data sets targeting multiple languages and prediction task types, to compare the proposed model with traditional, state-of-the-art, and other neural network strategies.

1 Introduction

For decades, readability assessment has been used by diverse stakeholders–from educators to public institutions—for determining the complexity of texts (Benjamin, 2012). Traditional formulas do so by focusing only on superficial linguistic features (e.g., average length of sentences or syllables per word). This leads to criticism, as these formulas do not explore deeper levels of text processing and thus yield rough estimates of complexity (i.e., difficulty) that often lack accuracy (Arfé et al., 2018). In fact, traditional formulas can label a text as ‘‘easy to read’’ even if its content is completely nonsensical (Davison and Kantor, 1982).

To improve the quality of automatic readability assessment, researchers turned to more sophisticated techniques that go beyond examining shallow features. These techniques, typically based on supervised machine learning, incorporate hundreds (even thousands) of features that describe a text from multiple perspectives: syntax, morphology, cohesion, discourse structure, and subject matter (Dell’Orletta et al., 2011; François and Fairon, 2012; Denning et al., 2016; Arfé et al., 2018). The dependency on these numerous features, however, has made readability assessment tools too complex to deploy and apply to languages beyond the one for which they were originally designed. Furthermore, feature and language dependency, along with lack of homogeneity in terms of readability scales, often prevent researchers from comparing new strategies with state-of-the-art counterparts, preventing community consensus on which features are the most beneficial for capturing text complexity (De Clercq and Hoste, 2016).

Existing literature reflects the fact that applications that leverage text complexity analysis, including book recommendation or categorization (Lexile, 2016; Pera and Ng, 2014), Web result summarization (Kanungo and Orr, 2009), and accessibility in the health domain (Bernstam et al., 2005; Fitzsimmons et al., 2010), still favor less precise but easier to implement alternatives, with Flesch as the most accepted choice (Ballesteros-Peña and Fernández-Aedo, 2013; Bea-Muñoz et al., 2015). We argue that this is caused by the uncertainty induced by the lack of uniformity of readability scales, adaptability among readability assessment tools, and benchmarks.

Areas of study that were historically heavily dependent on feature engineering, including sentiment analysis or image processing (Manjunath and Ma, 1996; Abbasi et al., 2008), have made their way towards alternatives that do not involve manually developing features, and instead favor deep learning (Wang et al., 2016). This resulted in more reproducible strategies—easily portable to other domains or languages, as they only require implementing the *structure* of a specific neural network and just rely on *core components* of resources, such as words, signals, or pixels, rather than features specifically designed for a domain or language.

Transactions of the Association for Computational Linguistics, vol. 7, pp. 421–436, 2019. https://doi.org/10.1162/tacl_a_00278.
Action Editor: Stephen Clark. Submission batch: 2/2019; Revision batch: 5/2019; Published 7/2019.

Issues pertaining to readability assessment are not limited to performance and adaptability. As stated by Benjamin (2012), a teacher should never use a readability score blindly when giving a text to a student, as specifics of the difficulties of the reader and the text should always be considered in this process. For this pairing to be successful, it is imperative for readability assessment tools to provide information beyond a single score. The explainability issue has been addressed in systems like Coh-Metrix (Graesser et al., 2011) by showing users the individual values of the features incorporated in the system. This strategy, however, has been criticized by the education community as most features presented are not straightforward to understand for people without background in both computation and linguistics (Elfenbein, 2011). More intuitive explanations could greatly ease the use of readability tools.

In this paper, we present a multilingual automatic readability assessment strategy based on deep learning: **Vec2Read**.[1] We still follow the premise of words being the core components for a neural network that deals with text. However, in order to avoid the aforementioned domain dependency issue and adapt the architecture to the readability task, we inform our model with part of speech (POS) and morphological tags. This is done by a *multiattentive structure* that allows the network to filter important words that influence the final complexity level estimation of a text. Apart from informing the network, the multiattentive structure can also be used to offer users further insights on which parts of a text have the most influence for determining its reading level.

Our research contributions include the following:

- We propose a multiattentive recurrent deep learning architecture specifically oriented to the readability assessment task.
- The proposed strategy is, to the best of our knowledge, the first capable of estimating readability in more than two languages.
- We incorporate an attention structure that allows a model to use multiple focuses of attention (with different degrees of importance) to inform word selection.
- We conduct an exhaustive evaluation based on different languages, readability-measuring scales, and data sets of varied sizes, in order to compare the performance of Vec2Read with existing baselines, a comparison that is rarely done in this area due to lack of benchmarks.
- We present an initial analysis on the use of attention mechanisms as a potential alternative for providing explanations for readability.

Task Definition. Given a text t, use model M to predict its reading level. The functionality of M is directly dependent on the characteristics of a data set D used for training: *language* and *readability scale*. The scale can be discrete (binary or multilevel) or continuous. Any language is viable; for data set availability we train M for Basque, Catalan, Dutch, English, French, Italian, and Spanish.

2 Method

In this section we introduce Vec2Read, a multiattentive recurrent neural network architecture for readability assessment.

2.1 General Architecture

The general architecture of Vec2Read (illustrated in Figure 1) is designed to emulate the structure of a text. A text is inherently **recurrent**, as it is composed of a series of words that depend on each other in order to produce a message. A text is also **hierarchical**, as it is composed of structural components such as sentences or paragraphs in order to group information.Vec2Read takes into account both characteristics to better capture text structure. Unlike existing hierarchical neural networks that take advantage of both word and sentence level recurrent layers (Yang et al., 2016), Vec2Read has a single recurrent layer at word-level; hierarchical information is used to generate both word- and sentence-level attention scores for creating a text representation.

2.2 Input

Given a text t, let the input of Vec2Read be $x = < x_w, x_p, x_m >$, where x_w, x_p, and x_m represent data structures containing a sequence of tokens in t, their corresponding POS tags, and morphological tags, respectively. x_{w_i} refers to the i^{th} sentence in t and $x_{w_{ij}}$ is the j^{th} token in

[1]The implementation and evaluation framework code is available on a public repository: https://github.com/ionmadrazo/Vec2Read.

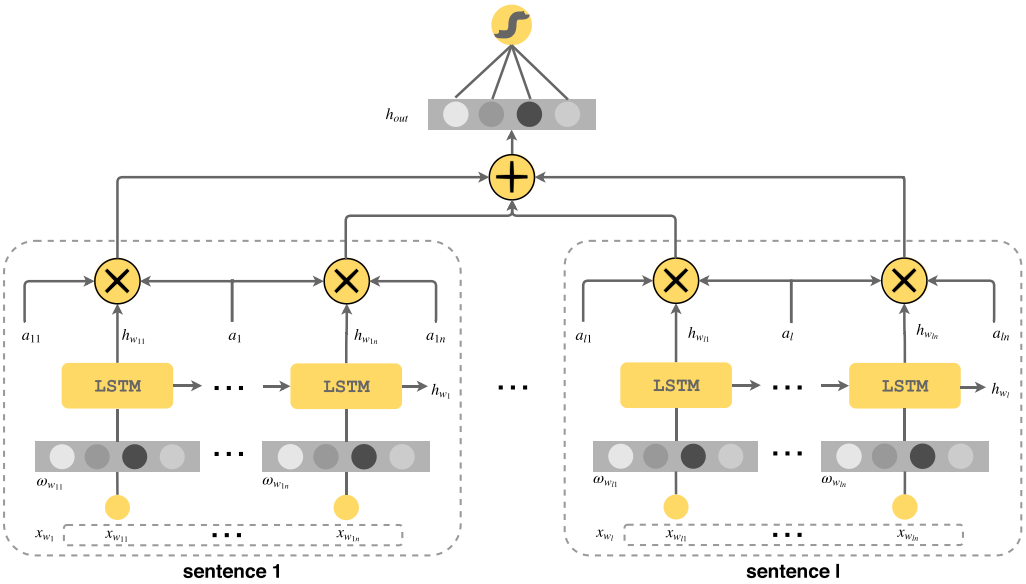

Figure 1: Description of the general architecture of Vec2Read.

x_{w_i}. x_{p_i} and x_{m_i} refer to the POS and morphological tag sequences for x_{w_i}, and $x_{p_{ij}}$ and $x_{m_{ij}}$ represent the POS and the morphological tags for $x_{w_{ij}}$. Note that $x_{m_{ij}}$ contains a set of tags per word rather than a single token or POS label. For instance, given the word *plays*: $x_{w_{ij}} =$ *''plays''*, $x_{p_{ij}} =$ *''Verb''*, and $x_{m_{ij}} =$ *''{Tense: present, Person: 3...}''*. To ease further processing, $x_{m_{ij}}$ always contains all possible morphological tags considered for the language, assigning a *Not applicable (NA)* value when the label cannot be applied to the token—for example, tense would have a value of *NA* for all nouns. The number of tags used is language dependent. (See Section 3.1 for details on tag set used in the experiments.)

2.3 Dense Vector Representations

Dense vector representations or embeddings have shown to be useful for representing discrete values, such as words, in applications dealing with text (Tang et al., 2014; Madrazo Azpiazu et al., 2018). Vec2Read converts all discrete values in x into dense vector representations before feeding them to the model. This is achieved by using a lookup table $\Omega_w \in \mathbb{R}^{v \times d}$ where each row is an embedding for a specific word in the vocabulary, v is the vocabulary size, and d is the number of latent features used for representation. Similarly, lookup tables Ω_p and Ω_m are used for representing POS and morphological tags, respectively. $\omega_{w_{ij}}$ refers to the embedding of $x_{w_{ij}}$; $\omega_{p_{ij}}$ to the embedding of the POS tag of $x_{w_{ij}}$; and $\omega_{m_{ij}}$ to the embedding that captures the morphological information of $x_{w_{ij}}$ created by concatenating the representations of each morphological tag in $x_{m_{ij}}$. Ω_w, Ω_p, and Ω_m can be either initialized using random uniform distributions and then trained along with the other weights of our model or based on pretrained representations (see Section 3.1). Note that representations of each input type are maintained separately and can therefore be of different size.

2.4 Encoding Sentences and Words

A recurrent neural network (RNN) (Grossberg, 1988) is an extension of a traditional neural network where each node in a layer takes as input not only information from the previous layer but also from a node in the same layer located directly next to it. This creates a structure designed to handle sequences like words in a text. Unfortunately, traditional RNNs are prone to the vanishing gradient problem that makes them difficult to train, hindering final performance (Hochreiter, 1991). A long short-term memory (LSTM) network (Hochreiter and Schmidhuber, 1997) addresses a traditional RNN's vanishing gradient problem by using several gates on each RNN cell responsible for storing or forgetting information from the cell state.

Vec2Read uses a bidirectional LSTM network that considers the input sentences in forward and backward directions for creating representations of whole sentences and individual words. We refer to h_{w_i} as the representation of x_{w_i}, obtained by concatenating the outputs of the final states of

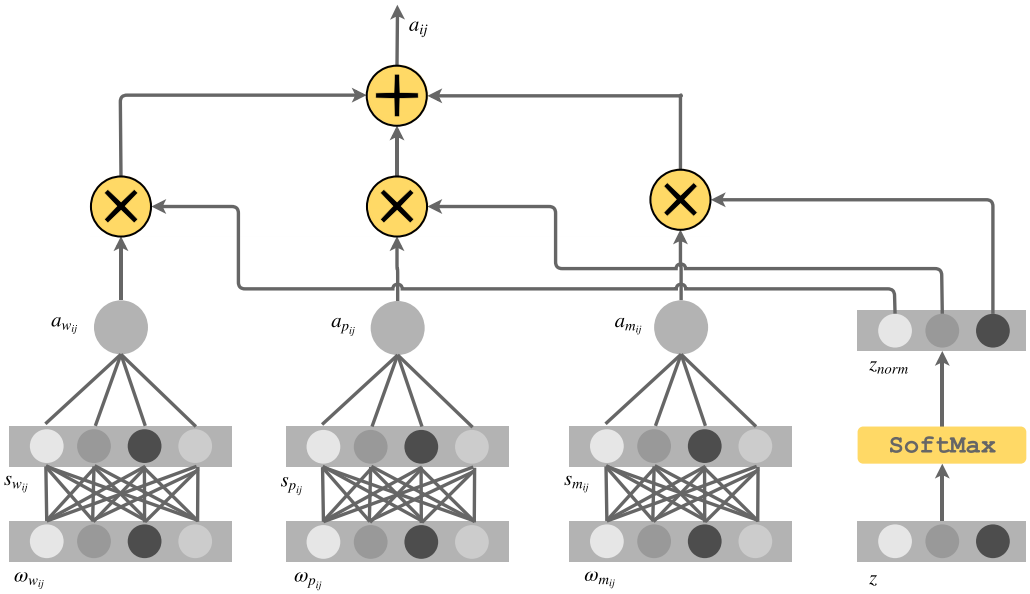

Figure 2: Description of the multiattentive network for token in position j in sentence i.

the LSTM network in both the forward and final pass; $h_{w_{ij}}$ is the representation generated by the LSTM network at time step j (i.e., for word $x_{w_{ij}}$) for i, concatenating the outputs of forward and backward passes.

2.5 Textual Representation Layer

A final general representation of t, denoted h_{out}, is created by aggregating all the encoded word representations generated by the LSTM network (Equation 1). This is done using a weighted sum over $h_{w_{ij}}$, where the weights are defined by the attention mechanism described in Section 2.6.

$$h_{out} = \frac{\sum_{i=1}^{l} \sum_{j=1}^{n_i} a_i a_{ij} h_{w_{ij}}}{\sum_{i=1}^{l} n_i} \quad (1)$$

where a_i is the attention generated for sentence i, a_{ij} is the attention for $x_{w_{ij}}$, n_i reflects the number of tokens in sentence i, and l is the number of sentences in t. The denominator is a normalization factor meant to remove the effect of length in texts. This normalization factor is especially important for readability prediction, given that the network could otherwise learn to discriminate texts based mostly on length, due to a strong bias in readability data sets for harder texts to be longer. Informing the model with length distribution of texts in each reading level could lead to performance improvement in an experimental setting. However, doing so would not allow us to estimate model performance in a real scenario, where text length will rarely follow the distribution seen in training sets. Therefore, we favor a length-independent model.

2.6 Attention Mechanism

Vec2Read is designed to capture the general structure of t in order to predict its reading level. Although one could argue that the reading level of a text is dependent on every one of its words, text simplification studies (Glavaš and Štajner, 2015; Paetzold and Specia, 2016) indicate that difficulty is generally introduced in a text by specific words and sentences—just a few hard sentences could significantly increase overall text difficulty. Following this intuition, Vec2Read uses an attention-generation mechanism (described in Figure 2) capable of predicting which parts of t have the most influence in its overall difficult. This way, our model can focus on the important parts of t and provide a more accurate readability estimation.

The attention mechanism of Vec2Read works on two levels: sentence and word. It detects which sentences have most influence towards determining the reading level of t and also which words are most influential. Each of these two-level predictions are composed of three attentions, oriented to consider the influence of each part of t from three linguistic perspectives: semantic, syntactic, and morphologic.

We now describe how the multiattentive mechanism works at word level, then we detail

how to adapt this model for the sentence-level version.

2.6.1 Word-Level Attention

The word level attention mechanism consists of three single attention mechanisms that are aggregated. Each individual attention network follows the same structure, a two-layer neural network, only differing on the size of the input and the number of hidden units. We set the number of hidden units proportional to the input length (see Section 3.1 for configuration details). Specifically, we compute each attention score $a_{att_{ij}}$ as follows:

$$\begin{aligned} s_{att_{ij}} &= \sigma(W_{att} \times \omega_{att_{ij}} + b_{att}) \\ a_{att_{ij}} &= \sigma(W_{att2} \times s_{att_{ij}} + b_{att2}) \end{aligned} \quad (2)$$

where $att \in \{w, p, m\}$ is an attention type, W_{att} and W_{att2} are the weights of the first and second network layers, b_{att} and b_{att2} are their respective biases, $s_{att_{ij}}$ is an intermediary representation, and σ is a sigmoid activation function.

Similar to the model in Figure 1, the input for generating semantic and syntactic attention scores are $\omega_{w_{ij}}$ and $\omega_{p_{ij}}$. For calculating morphological attention scores, the input is instead the concatenation of each of the morphological tag embeddings in $\omega_{m_{ij}}$.

After generating a score using each single attention mechanism, Vec2Read **aggregates** them into one value that will be the final attention score predicted for $x_{w_{ij}}$. Previous works in feature engineering for readability assessment indicate that not all features are of equal importance for predicting the readability of a text (Dell'Orletta et al., 2011; Gonzalez-Dios et al., 2014). We believe that this phenomena also apply to attention generation, and therefore each single attention will not contribute equally to the final attention prediction.

To allow our model the flexibility of deciding which attention matters most, we use an attention aggregation strategy that assigns a different weight to each attention. $z = < z_w, z_p, z_m >$ is a vector containing the weights corresponding to each attention mechanism, which are automatically estimated during the training phase to allow Vec2Read to learn which attention has the most influence. We constrain the weights to sum to 1 by applying a softmax function to z:

$$z_{norm_{att}} = \frac{\exp(z_{att})}{\sum_{att} \exp(z_{att})} \quad (3)$$

The final attention a_{ij} for $x_{w_{ij}}$ is calculated as:

$$a_{ij} = \sum_{att} z_{norm_{att}} \times a_{att_{ij}} \quad (4)$$

Lastly, we constrain all word attentions in a sentence to sum to 1 using a softmax function.

2.6.2 Sentence Level Attention

Sentence level attention follows the same structure as word level attention described in Section 2.6.1, differing only on how the inputs of each single attention network are generated. In this case, for the semantic attention we use h_{w_i} vectors already defined in the general architecture (see Figure 1); for syntactic and morphological attentions we feed separate LSTM models using the sequence of syntactic and morphological embeddings in the sentence and use the output of the last recurrent step as input to the attention mechanism. We then normalize sentence level attentions so that they sum to one using a softmax function.

2.7 Output Layer

The output layer of Vec2Read is responsible for mapping h_{out} to a reading level prediction. Two different output layers are used depending on the type of prediction required in each task: discrete or continuous.

2.7.1 Discrete Prediction

To predict a discrete reading level for t, Vec2Read generates a probability distribution over each reading level $\hat{y} \in [0, 1]^c$, where c represents the set of possible prediction classes, that is, reading levels. This is achieved by applying a fully connected layer with a softmax activation function to h_{out} to ensure that the probabilities in $\hat{y}$ add up to one.

$$\hat{y} = \text{softmax}(W_{out} \times h_{out}^{\top} + b_{out}) \quad (5)$$

where $W_{out} \in \mathbb{R}^{|c| \times r}$ is the matrix of weights of the fully connected layer, b_{out} is a vector of length $|c|$ containing the biases, $|c|$ is the number of possible reading levels to be predicted, r is the number of latent features in h_{out}, and $\top$ refers to the transpose operation. The class that yields the highest probability is the one assigned to t.

2.7.2 Continuous Prediction

When the reading level of t is defined as a continuous value, Vec2Read generates a real value $\hat{y} \in [y_{min}, y_{max}]$, where y_{min} and y_{max} refer

to the minimum and maximum readability score possible in the used scale. This is achieved by applying a fully connected layer with a min-max leaky rectified linear unit as activation function. The leaky version of this function is favored given its benefits in terms of avoiding neuron death during training (Xu et al., 2015a).

$$\hat{y} = \vartheta(W_{out} \times h_{out}^{\top} + b_{out}) \tag{6}$$

$$\vartheta(q) = \begin{cases} y_{max} + \varepsilon * q, & q > y_{max} \\ q, & y_{min} < q < y_{max} \\ y_{min} - \varepsilon * q, & q < y_{min} \end{cases} \tag{7}$$

where $W_{out} \in \mathbb{R}^{1 \times r}$ is the matrix of weights of the fully connected layer, b_{out} is a bias, r is the number of latent features in h_{out}, $\top$ refers to the transpose operation, and ε is a constant set to 0.001 during training and to 0 during prediction.

2.8 Fitting Parameters

For fitting the parameters of our model we use stochastic gradient descent. This strategy computes the prediction of our model given specific data, and compares it to the actual objective value using an error or loss function. The goal is to minimize the error for which a gradient is backpropagated to each of the parameters in the model by subsequently updating them in a direction that will minimize the overall prediction error. As the objective function for training the model, we consider two different loss functions, depending on how the reading level is estimated.

For **discrete** predictions, we used cross-entropy:

$$H(y, \hat{y}) = -\sum_{i=1}^{|c|} y_i \log(\hat{y}_i) \tag{8}$$

where $\hat{y} = < \hat{y}_1, .., \hat{y}_{|c|} >$ is the probability distribution predicted by our model and $y = < y_1, .., y_{|c|} >$ is the one-hot encoded vector representing the target class.

For **continuous** predictions, we use instead mean square error (MSE):

$$\text{MSE} = \frac{1}{|D|} \sum_{d \in D} (\hat{y}_d - y_d)^2 \tag{9}$$

where D is a collection of texts in a given data set, $|D|$ is the number of documents in D, and $\hat{y}_d$ and y_d are the prediction generated by our model for document d and its ground-truth, respectively.

3 Experiments and Discussion

In this section, we first describe model configuration. We then outline data sets and baselines considered for evaluation purposes. Lastly, we discuss the results of the analysis conducted to verify the overall performance of Vec2Read and showcase the validity of its attention mechanism.

3.1 Model Setup

We describe Vec2Read's configuration; parameters were empirically determined using a hold-out set as escribed in Section 3.4.

Optimization. For fitting the parameters of our model, we used the Adaptive Movement Estimation (Kingma and Ba, 2014); learning rate = 0.001.

Initializations. For Ω_w we used a pretrained version of word embeddings, which were trained using a skip-gram algorithm on Wikipedia documents, as described in Bojanowski et al. (2017). All the remaining weights and biases of our model, as well as initial states of LSTM layers, were initialized using a random uniform distribution.

Dimensions. The number of hidden units in the semantic, syntactic, and morphologic LSTM networks were empirically set to 128, 32, 64, respectively. The dimensions of the embedding representations were set to 300, 16, 16. Given that the input of the morphological attention combines multiple embeddings corresponding to the morphological labels used, the final dimension of $\omega_{m_{ij}}$ is $u \times 16$, where u is the number of tags used.

Tagging. We used SyntaxNet (Andor et al., 2016) trained on Universal Dependencies data sets v1.3 for computing the POS and Morphology tags of words. All POS and Morphology tags available in the data set were used. Accuracy per language is varied, Dutch being the one with lowest accuracy (POS: 89.89%, Morph: 89.12%) and Catalan the language where the tagging is most accurate (POS: 98.06%, Morph: 97.56%).[2]

3.2 Data Sets

For assessment and analysis purposes, we use several data sets based on both expert-labeled

[2]For per language accuracy details see `https://github.com/mldbai/tensorflow-models/blob/master/syntaxnet/universal.md`.

	SimpleWiki		Wizenoze					Newsela							
	S	C	1	2	3	4	5	3	4	5	6	7	8	9	12
Words per text	111	5987	35	128	67	266	801	448	674	777	872	927	990	970	1169
Sentences per text	6	222	3	10	5	16	35	43	54	54	54	52	46	46	50
Syllables per word	1.31	1.37	1.40	1.41	1.44	1.52	1.53	1.27	1.30	1.33	1.36	1.39	1.40	1.43	1.42
Words per sentence	17	25	11	14	14	16	21	10	12	14	16	18	20	21	24
Ratio of unique words	0.69	0.32	0.86	0.79	0.79	0.65	0.55	0.44	0.42	0.42	0.42	0.43	0.43	0.43	0.43
Flesch-Kincaid	6.37	10.70	5.40	6.61	6.65	8.65	10.76	3.42	4.63	5.72	6.81	7.77	8.73	9.68	10.48

	WikiViki		Ikasbil					MTDE*			
	S	C	A2	B1	B2	C1	C2	1	2	3	4
Words per text	303	6036	215	276	320	327	354	294	276	288	301
Sentences per text	16	217	21	18	18	16	15	11	12	13	23
Syllables per word	1.36	1.38	1.40	1.39	1.43	1.37	1.40	1.51	1.47	1.37	1.23
Words per sentence	17	25	10	15	17	20	23	26	23	23	14
Ratio of unique words	0.62	0.31	0.51	0.53	0.52	0.52	0.51	0.56	0.56	0.54	0.49
Flesch-Kincaid	7.13	10.71	4.83	6.66	7.91	8.38	9.90	12.5	10.71	9.63	4.64

Table 1: Statistics on the data sets considered in our assessment, where S and C stand for Simple and Complex, respectively. When data sets are multilingual, texts from all languages are considered for computing average. * Given that ground truth scores for MTDE are continuous, for illustration purposes we reported statistics grouped in 4 levels, i.e., 0-25, 26-50, 51-75, 76-100 (original values preserved in the experiments).

educational materials (Ikasbil, Newsela, Wizenoze) and crowd-source generated and simplified texts (MTDE, SimpleWiki, VikiWiki). We describe each data set below; detailed statistics are in Table 1.

SimpleWiki. Simple.Wikipedia(.org) is a simplified version of the most representative articles in English Wikipedia written with simple vocabulary and grammar. These articles target readers who are learning English. We created a binary (simple or complex) data set using the 131,459 articles available in Simple Wikipedia and their Wikipedia counterparts, totaling 262,918 documents. The use of Simple.Wikipidia/Wikipedia articles has already proved to be useful for readability and simplification assessment (Ambati et al., 2016), a fact we confirm in our qualitative analysis in Section 3.6. (See Wikimedia [2018] for details on how articles on Simple.Wikipedia are simplified.)

VikiWiki. Vikidia(.org) is similar to Simple Wikipedia, but it is not constrained to articles written in English. Following a similar procedure to SimpleWiki, we created VikiWiki using all the articles in Vikidia along with their Wikipedia counterparts. The data set comprises 70,514 documents: 23,648 in French, 9,470 in Italian, 8,390 in Spanish, 3,534 in English, 924 in Catalan, and 898 in Basque, uniformly distributed among simple and complex levels.

MTDE. MTDE is the data set presented in De Clercq and Hoste (2016), generated using crowd-sourcing techniques. It consists of 105 documents both in English and Dutch, each labeled with a score in the 0–100 range that indicates its complexity.

Newsela. Newsela is an instructional content platform that provides reading materials for classroom use. As part of their research program, Newsela makes available a sample of their labeled corpora, which we use for evaluation. The data set consists of 10,786 documents distributed among grade levels 2–12 (around 1,200 per level for English and 120 for Spanish). We excluded from our experiments grade levels 2, 10, and 11, as the number documents for those levels are significantly lower when compared with other classes (284, 11, and 2, respectively, for English).

Ikasbil. Ikasbil (2018) is an online resource for learning Basque containing articles leveled following the Common European Framework of Reference for Languages. Using this source, we created a data set consisting of *5 reading levels* (A2, B1, B2, C1, and C2), with 200 documents per level. Level A1 was omitted due to insufficient documents.

Wizenoze. Data set provided by Wizenoze (2018), an online platform dedicated to easing the retrieval of (curated) resources suitable for the classroom setting. The data set consists of

2,000 documents in English and Dutch, equally distributed and labeled using a 5-level readability scale (1–5).

3.3 Compared Strategies

We now describe the strategies considered in our assessment, including traditional formulas, state-of-the-art tools based on extensive feature engineering, and neural network structures intended for an ablation study on major components of Vec2Read.

3.3.1 Traditional Strategies

Flesch. Even if simple, Flesch (1948) remains one of the most used readability formulas and is therefore treated as a baseline by authors of publications pertaining to readability estimation. In addition to the traditional version for English texts, we consider language-specific adaptations (Kandel and Moles, 1958; Fernández Huerta, 1959; Douma, 1960; Lucisano and Piemontese, 1988). We followed the framework used in Madrazo Azpiazu (2017), which maps the Flesch score of a given text t into a binary value (simple or complex) based on its distance with the average Flesch score computed using the training documents for the respective classes.

3.3.2 State-of-the-Art Strategies

S1. The system proposed by De Clercq and Hoste (2016) is the only one designed for readability assessment for more than one language: Dutch and English. Its design consists of a support vector machine that uses ad hoc features to capture varied linguistic characteristics of texts (e.g., syntax or semantics). Given that the algorithm implementation is not publicly available, comparisons against this strategy are based on results reported in De Clercq and Hoste (2016).

S2. A multilevel Basque readability assessment strategy that relies on random forest and linguistic features with a major emphasis on morphology and syntax (Madrazo, 2014). The authors provided their data set (including cross-validation folds) for comparison purposes. Because of lack of implementation availability, comparisons against S2 are limited to the Basque language.

S3. Similar to S2, the strategy introduced in Madrazo Azpiazu (2017) also relies on a random Forest and linguistic features. Given implementation availability, we adapted it to run on all discrete and continuous prediction tasks by changing its linguistic annotation tools. For fairness in the comparison, we used the same linguistic annotation tools used by Vec2Read (described in Section 3.1).

S1, S2, and S3 are treated as examples of feature engineered state-of-the-art strategies.

3.3.3 Ablation Study Strategies

To determine the utility of each feature incorporated in the architecture of Vec2Read, we consider several variations of Vec2Read in the assessment.

FC. A two-layer fully connected neural network with 256 hidden units, taking as input the average of the word embeddings of all words in a text.

¬Attention. Basic architecture of Vec2Read. It maintains Vec2Read's hierarchical and recurrent structure, but overrides the output of its attention generation mechanism by assigning each word and sentence a uniformly distributed attention score.

¬Word, **¬Sent**, **¬Sem**, **¬Syn**, and **¬Morph**. Vec2Read architecture without word-level, sentence-level, semantic, syntactic, and morphological attention, respectively.

3.4 Experimental Setup

We followed a 10-cross-fold validation framework for measuring the performance of each strategy considered. A disjoint stratified 10% of data in SimpleWiki (includes both simple and complex) was excluded from the experiments and used for developmental and hyper-parameter tuning purposes. Note that to abide by the adaptability premise intended for our model, we only tuned hyper-parameters for English. Doing so allows us to understand to what extent the model can directly transfer to other languages without language-specific tuning, thus simulating a real-world scenario for tool adaptation.

To conduct fair comparisons, we used the same cross-validation folds across experiments (when possible, we used the folds made publicly available; otherwise we re-run strategies using our data and folds). The only exception are experiments related to S1, for which we could only access the original data set. Consequently, we compare our results with respect to those published in De Clercq and Hoste (2016).

Data set	Lang.	Flesch	State of the art			Ablation							Vec2Read
			S1	S2	S3	FC	¬Attention	¬Word	¬Sent	¬Sem	¬Syn	¬Morph	
Binary Prediction (Accuracy)													
SimpleWiki	en	.724	-	-	.822	.722	.877	.893	.896	.887	.897	.915	**.918**⋆
VikiWiki	en	.720	-	-	.827	.721	.852	.860	.862	.859	.868	.876	**.879**⋆
	es	.687	-	-	.792	.719	.816	.823	.831	.828	.835	.839	**.847**⋆
	fr	.670	-	-	.842	.756	.864	.869	.870	.869	.870	.872	**.884**⋆
	it	.653	-	-	.755	.766	.783	.797	.802	.793	.801	.805	**.814**⋆
	eu	-	-	-	**.693**	.648	.682	.683	.686	.684	.684	.685	.687
	ca	-	-	-	.733	.677	.715	.725	.737	.728	.732	.734	**.742**
Multilevel Prediction (Accuracy)													
Ikasbil	eu	-	-	.625	.622	.617	.679	.685	.689	.681	.684	.686	**.692**⋆
Newsela	en	-	-	-	.464	.447	.489	.501	.517	.498	.502	.525	**.527**⋆
	es	-	-	-	.467	.452	.487	.494	.510	.504	.509	.503	**.519**⋆
Wizenoze	en	-	-	-	.649	.631	.665	.678	.685	.682	.685	.700	**.701**⋆
	du	-	-	-	.652	.636	.668	.679	.687	.681	6.85	.683	**.695**⋆
Continuous Prediction (RMSE)													
MTDE	du	-	**.0003**⋆	-	-	.0171	.0068	.0064	.0064	.0066	.0062	.0059	.0059
	en	-	.0060	-	-	.0184	.0054	.0052	.0051	.0051	.0053	.0051	**.0051**

Table 2: Performance comparison among traditional, state-of-the-art, ablation strategies, and Vec2Read on different data sets. '*' denotes statistically significant improvement over counterparts (Flesch, S1, S2, S3, Vec2Read). Accuracy (higher is better) is reported for all data sets except for MTDE, where RMSE (lower is better) is used in order to be able to compare with S1. Cells marked with '-' denote that the strategy is not applicable to the data set.

3.5 Overall Performance

As mentioned by De Clercq and Hoste (2016), each work in the readability area interprets the readability estimation task in a different manner—using different languages and data sets—often making the community unable to compare proposed tools with each other. In order to best contextualize the performance of Vec2Read, we consider a broad set of tasks using data sets of varied (i) **size**, that go from 105 documents to 262,918, (ii) **language**, considering seven languages, and (iii) **prediction type**, namely, binary, multilevel, and continuous predictions.

To quantify performance of different readability estimation alternatives, we use *accuracy* for classification tasks and *Root Mean Square Error* (RMSE) for regression tasks. Table 2 summarizes the results obtained by Vec2Read and its counterparts on the aforementioned data sets. As we followed a 10-cross-fold validation framework, scores in Table 2 correspond to the averages over the 10 folds. Statistical significance was tested using a paired t-test with a confidence interval of $p < 0.05$.

General Discussion. As anticipated, we observe that traditional formulas (Flesch) yield the lowest performance, followed by the general-purpose neural network approach (FC). This validates our hypothesis that a neural network that simply considers words without considering text structure or other linguistic features is not enough for readability assessment. Further, models that consider richer traits of text, such as Vec2Read and its attention-less version (¬*Attention*), are consistently comparable or outperform state-of-the-art strategies (S1, S2, S3), demonstrating the validity of the proposed architecture. Vec2Read achieved a statistically lower rate only for 1 out of 14 tasks (defined as a data set–language pair) in our evaluation. We attribute this to the size of the data set, which only includes 105 texts. It is anticipated for a strategy based on feature engineering such as S1, which has been specifically designed for Dutch, to outperform a neural network based counterpart (such as Vec2Read), as the latter is known to need large amounts of data for best performance.

Data set size. The number of instances used for training has a strong effect on the overall performance of Vec2Read. All the analyzed strategies generate lower scores for smaller data sets; performance drop is more prominent among the strategies based on deep learning (Vec2Read and all the ablation strategies). We attribute this behavior to the higher variance of deep learning

Data set	Lang	Words	Part Of Speech	Morphological
SimpleWiki	en	unincorporated, reside, inhabitants	CCONJ, SCONJ, DET	Relative (pronoun), Past, Infinitive
VikiWiki	en	belonged, abolished, comprising	SCONJ, CCONJ, DET	Relative (pronoun), Infinitive, Past
	es	recae, mantiene, consiste	SCONJ, ADJ, AUX	Participle, Subjunctive, Past
	fr	circonscriptions, associer, comporter	CCONJ, ADV, SCONJ	Reflexive, Subjunctive, Passive
	it	comprende, risiede, rivelato	CCONJ, SCONJ, VERB	Past, Subjunctive, Relative (pronoun)
	eu	aldarrikapen, gizarte, eskumen	NOUN, ADJ, DET	Subjunctive, Inessive, Dative
	ca	acreditat, mantenir, contribuint	ADJ, NOUN, CCONJ	Subjunctive, Relative (pronoun), Participle
Ikasbil	eu	hedatu, irudikatu, biltzartu	CCONJ, VERB, SCONJ	Subjunctive, Genitive, Inessive
MTDE	du	geregeld, omvat, stemhebbend	NOUN, ADJ, CCONJ	Past, Participle, Infinitive
	en	handled, retained, consisting	NOUN, SCONJ, ADJ	3rd Person, Relative (pronoun), Past
Newsela	en	aquaponics, government, unwavering	CCONJ, SCONJ, ADJ	Infinitive, Relative (pronoun), Past
	es	postularse, extintos, realizacion	CCONJ, SCONJ, AUX	Subjunctive, 3rd Person, Participle
Wizenoze	en	controversy, transition, equality	SCONJ, CONJ, NOUN	Relative (pronoun), 3rd Person, Past
	du	vervaardiging, afgezette, bijgevolg	CCONJ, NOUN, ADJ	Participle, Past, Infinitive

Table 3: Words, POS tags, and Morphological tags that receive highest attention from Vec2Read. [3]

models, needing more data than feature engineered models to achieve good generalization. In addition, we also note that the attention mechanism becomes more useful the larger the data set and its effect is negligible in small data sets such as MTDE.

Language and task type. We observe no emerging patterns in terms of performance induced by the language or the type of task. One could argue that results for English are in general higher, although we attribute these differences to data set size (English data sets are in general larger) rather than to the language itself. Accuracy scores for multilevel estimation are lower than for binary, which is expected, as it is harder for a model to learn readability predictions for scales that go beyond just simple or complex.

Ablation study. By comparing Vec2Read with its attention-less counterpart (¬*Attention*) we can conclude that the proposed multiattentive mechanism has indeed a positive effect for readability prediction. In 11 out of 14 tasks the multiattentive mechanism achieved statistically significant improvements over ¬*Attention*; for the remaining 3 tasks (VikiWiki-EU, MTDE-EN, MTDE-DU) there was no statistically relevant difference. The usefulness of the attention mechanism is influenced by the size of the data set, as the larger the data set, the more prominent the improvement obtained by the model using the attention mechanism. We also notice that the difference of using the morphological attention for certain languages such as English is insignificant whereas it is more prominent in other languages, a fact we attribute to the low morphological diversity of English.

[3]For definition of POS tags refer to https://universaldependencies.org/u/pos/.

3.6 Attention Mechanism

As outlined in Section 3.5, the attention mechanism of Vec2Read leads to improvement in prediction performance. In this section, we aim to shed light on what the attention mechanism is actually learning to do and whether this information could be used for explaining the estimated reading levels from a more *qualitative* standpoint. Even if attention mechanisms are used in manifold applications, there exists no defined framework for evaluating their behavior. Instead, researchers focus on finding explanations of what the mechanism is learning (Hermann et al., 2015; Xu et al., 2015b). For this reason, the following discussion is not intended to be conclusive but instead to provide initial results meant to be inspirational for future work on readability prediction explainability.

In order to illustrate the parts of a text that receive the most attention from Vec2Read, we show in Table 3 the top-3 words, POS, and morphological tags that score the highest attention level for each individual task. We observe that words that receive most attention are in general words that are not frequently used by an average speaker, and therefore can present a challenge for the reader. We also observe that conjunctions (used for making sentences longer) are consistently among the most influential POS tags and that subjunctive mood, passive voice, and specific verb forms, such as infinite or participle, are considered important by our model. Both the use of conjunctions and passive voice align with features already found positive in the readability literature (François and Fairon, 2012; Gonzalez-Dios et al., 2014), leading us to infer that the attention mechanism is learning valid assumptions

for detecting which parts of a text are most influential for readability prediction.

One of the benefits of using a multiattentive mechanism rather than a traditional attention mechanism that considers all features at once is that the model can adapt and give more importance to specific datapoints depending on the task. In order to illustrate how Vec2Read takes advantage of this functionality, we show in Table 4 the weights[4] assigned by the attention mechanism for each task (i.e., z_{norm}). We observe that higher weight is assigned to semantics when the data set is large, whereas syntax is more relevant for smaller data sets. This behavior depicts the adaptability of our model, using more generalizable information, such as POS tags, when data is scarce and taking advantage of more fine grained information, such as words, when data is abundant. Weights for morphological and syntactic attention are similar for most of the tasks with the exception of English, where morphology receives a lower weight compared with other languages. We attribute this phenomena to English being a morphologically poor language.

Consider Figures 3 and 4, two examples of attentions generated by Vec2Read. Figures 3 showcases the combined attention scores a_{ij} predicted by Vec2Read for a text snippet extracted from the English Wikipedia document about Qatna. The model used for predicting the attentions was trained using the SimpleWiki data set. In this example, we see that Vec2Read mostly focuses on complex nouns and adjectives, and tends to ignore less informative words, such as determiners.

Figures 4 shows the attentions generated for a sentence in Spanish by Vec2Read trained using Spanish VikiWiki. This example is meant to illustrate the ‘‘extra" information that can be obtained from a multiattentive mechanism, not only by showing which of the words are important for estimating text difficulty, but also hinting about why they influence the process. As captured in Figure 4, the connector *Consequentemente* (Consequently) is most important from a syntactic perspective, whereas the sequence *fue cerrado* (was closed) is more important from a morphological standpoint.

Manual analysis of the attention scores lead us to identify which parts of a text the model is focusing on. This initial examination reveals that the model is indeed learning about linguistic patterns known to be important for defining the difficulty of a text as opposed to stylistic biases caused by how the data sets were generated. This also serves as an indication for the validity of using crowd-sourced data sets, such as SimpleWiki and VikiWiki, for training purposes.

We found many examples where the multiattentive mechanism yielded interesting outputs, however, we also found some deficiencies we would like to highlight. Even if connectors, like *Consequentemente*, were detected correctly by Vec2Read, other commonly used connectors, such as *sin embargo* (nevertheless) or *a pesar de ello* (nonetheless), were not detected correctly given their multi-word structure. This indicates that word level attentions might not be enough for some languages, thus demonstrating the need to consider more sophisticated structures such as dependency trees, as well as other syntactic and morphological features of the text, in the future.

4 Related Work

Literature on automatic readability assessment is rich, not only in the languages to which existing strategies can be applied, but also on the diversity of linguistic perspectives that have been explored (Benjamin, 2012; Arfé et al., 2018).

Feature engineering has been the main focus in the readability assessment area. Techniques that exploit *shallow features* (e.g., number of syllables per word and average sentence length) remain a prominent strategy for estimating complexity levels of texts in diverse languages (Flesch, 1948; Spaulding, 1956; Al-Ajlan et al., 2008) and show better prediction capabilities than more sophisticated features when considered individually (Feng et al., 2010). Language models have also been proved useful when determining the reading level of a text (Schwarm and Ostendorf, 2005). The use of features capturing the *syntax* of a text have been demonstrated to be of great importance, as illustrated by Karpov et al. (2014), who built a system that heavily relies on features based on POS tags and the syntactic dependency tree of a text. Structural features may not influence text complexity estimation for languages like Chinese, which is why some researchers favor analyzing *lexical representations* (i.e., term frequencies; Chen et al., 2011). Even if not for most

[4]Weights averaged across 10 folds, see Section 3.5.

	SimpleWiki	VikiWiki						Ikasbil	MTDE		Newsela		Wizenoze	
	en	en	es	fr	it	eu	ca	eu	en	du	en	es	en	du
Semantic	.72	.67	.53	.55	.62	.43	.39	.68	.38	.31	.62	.41	.39	.32
Syntactic	.20	.26	.26	.19	.21	.28	.41	.15	.32	.37	.30	.35	.32	.40
Morphological	.08	.07	.21	.26	.17	.29	.20	.17	.30	.32	.08	.24	.29	.28

Table 4: Weights learned by Vec2Read for each of the data sets considered.

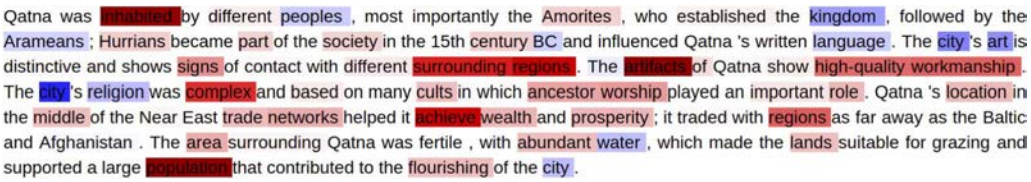

Figure 3: Attention scores generated by Vec2Read for a snippet of a Wikipedia article about Qatna. Color saturation indicates magnitude of the attention score, and hue indicates polarity (blue for simple, red for complex). Magnitudes are provided by the attention mechanism and the polarities are determined by the readability prediction generated when using each word as input to Vec2Read.

languages, *morphological* features have also been shown to be of great importance in terms of influencing the complexity level of texts written in languages known to be morphologically rich, such as Basque (Gonzalez-Dios et al., 2014). For considering *semantic* information in a text, existing works incorporate features related to true or false cognates, as a manner to better capture text difficulty for non-native readers (François and Fairon, 2012), or measure the coherence of the text based on graphical models (Mesgar and Strube, 2015, 2016, 2018). Unlike the aforementioned techniques, which rely on engineering features for specific languages and tasks, Vec2Read uses a deep learning strategy that automatically detects patterns related to readability.

Historically, readability assessment tools have been designed and evaluated in one language. To the best of our knowledge, only De Clercq and Hoste (2016) evaluate readability assessment performance in more than one language (i.e., Dutch and English) with the purpose of comparing the importance of features in each language. As presented in this paper, we go beyond two languages and instead quantify the performance of Vec2Read in seven different languages.

Attention mechanisms have been used with great success in several domains, including image classification (Xu et al., 2015b), question answering (Hermann et al., 2015), and automatic text translation (Bahdanau et al., 2014). The attention mechanism proposed for Vec2Read differs from the counterparts applied to the aforementioned tasks in the sense that it provides a composed attention score that can be decoupled to further analyze the influence individual words have in the overall complexity of a text from different linguistic perspectives.

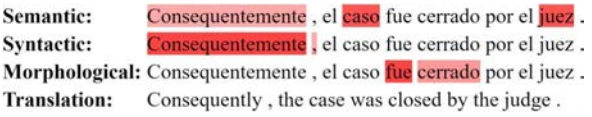

Figure 4: Scores generated using individual attention mechanisms by Vec2Read for a sentence in Spanish; saturation indicates the magnitude of the attention score.

5 Conclusion

We introduced Vec2Read, a multiattentive recurrent neural network architecture designed for automatic multilingual readability assessment. Vec2Read takes advantage of deep learning techniques by incorporating a multiattentive mechanism that allows the system to consider words and sentences that most influence the reading level of a text. We demonstrated the validity of our proposed architecture by conducting an exhaustive analysis using data sets in seven different languages and comparing Vec2Read to traditional, state-of-the-art, and other neural network architectures. Moreover, we outlined the benefits of this type of architecture for readability assessment, including the interpretability of the predictions using the attention scores.

This research work sets the foundations for language agnostic readability assessment, demonstrating that it is indeed possible to design a readability assessment strategy that works regardless of the language. This is achieved by disregarding hand-engineered features, historically known to be tedious to create and test, in favor of using simple tokens as input. We anticipate that given the magnitude and the diversity of the evaluation conducted, we have set a new baseline in the readability area, considerably harder to beat than the popularly used Flesch. This is supported by (i) the use of data sets in multiple languages that can, for the most part, be easily obtained and (ii) the release of our algorithm, so that other researchers can run it for comparison purposes. We expect this will make an area that is currently crowded with hard-to-compare systems finally progress towards more precise, usable, and comparable tools.

In the future, our research will be focused on generating more valuable explanations on what influences the readability of a text, as well as enhancing our model so that it can be trained jointly for multiple languages or can obtain benefit of cross-lingual data in order to improve the performance in languages with small corpora. We also plan on experimenting with character-based models, which could potentially take advantage of morphological information of texts without the need of a morphological tagger.

References

Ahmed Abbasi, Hsinchun Chen, and Arab Salem. 2008. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *Association of Computer Machinery Transactions on Information Systems (TOIS)*, 26(3):12.

Amani A Al-Ajlan, Hend S Al-Khalifa, and A Al-Salman. 2008. Towards the development of an automatic readability measurements for arabic language. In *Proceedings of the International Conference on Digital Information Management*, pages 506–511.

Bharat Ram Ambati, Siva Reddy, and Mark Steedman. 2016. Assessing relative sentence complexity using an incremental CCG parser. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1051–1057.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2442–2452.

Barbara Arfé, Lucia Mason, and Inmaculada Fajardo. 2018. Simplifying informational text structure for struggling readers. *Reading and Writing*, 31:2191–2210.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Sendoa Ballesteros-Peña and Irrintzi Fernández-Aedo. 2013. Análisis de la legibilidad lingüística de los prospectos de los medicamentos mediante el índice de flesch-szigriszt y la escala inflesz. *Anales del Sistema Sanitario de Navarra*, 36(3):397–406.

Manuel Bea-Muñoz, María Medina-Sánchez, and Mariano Tomas Flórez-García. 2015. Legibilidad de los documentos informativos en español dirigidos a lesionados medulares y accesibles por internet. *Anales del Sistema Sanitario de Navarra*, 38(2):255–262.

Rebekah George Benjamin. 2012. Reconstructing readability: Recent developments and recommendations in the analysis of text difficulty. *Educational Psychology Review*, 24(1):63–88.

Elmer V. Bernstam, Dawn M. Shelton, Muhammad Walji, and Funda Meric-Bernstam. 2005. Instruments to assess the quality of health information on the World Wide Web: What can our patients actually use? *International Journal Medical Information*, 74(1):13–19.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Yaw-Huei Chen, Yi-Han Tsai, and Yu-Ta Chen. 2011. Chinese readability assessment using tf-idf and svm. In *Proceedings of the International Conference on Machine Learning and Computing*, volume 2, pages 705–710.

Alice Davison and Robert N. Kantor. 1982. On the failure of readability formulas to define readable texts: A case study from adaptations. *Reading Research Quarterly*, 17:187–209.

Orphée De Clercq and Véronique Hoste. 2016. All mixed up? Finding the optimal feature set for general readability prediction and its application to english and dutch. *Computational Linguistics*, 42(3):457–490.

Felice Dell'Orletta, Simonetta Montemagni, and Giulia Venturi. 2011. Read-it: Assessing readability of italian texts with a view to text simplification. In *Workshop on Speech and Language Processing for Assistive Technologies*, pages 73–83.

Joel Denning, Maria Soledad Pera, and Yiu-Kai Ng. 2016. A readability level prediction tool for k-12 books. *Journal of the Association for Information Science and Technology*, 67(3):550–565.

W.H. Douma. 1960. De leesbaarheid van landblouwbladen een onderzoek naar en een toepassing van lees baarheidsformules. *afd. Sociologie en Sociographie van de Landbouwhogeschool te Wageningen*, Bulletin 17.

Andrew Elfenbein. 2011. Research in text and the uses of coh-metrix. *Educational Researcher*, 40(5):246–248.

Lijun Feng, Martin Jansche, Matt Huenerfauth, and Noémie Elhadad. 2010. A comparison of features for automatic readability assessment. In *Proceedings of the International Conference on Computational Linguistics: Posters*, pages 276–284.

José Fernández Huerta. 1959. Medidas sencillas de lecturabilidad. *Consigna*, 214:29–32.

P. R. Fitzsimmons, B. D. Michael, J. L. Hulley, and G. O. Scott. 2010. A readability assessment of online parkinson's disease information. *The Journal of the Royal College of Physicians of Edinburgh*, 40(4):292–296.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221.

Thomas François and Cédrick Fairon. 2012. An ai readability formula for french as a foreign language. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 466–477.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 63–68.

Itziar Gonzalez-Dios, María Jesús Aranzabe, Arantza Díaz de Ilarraza, and Haritz Salaberri. 2014. Simple or complex? Assessing the readability of Basque texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 334–344.

Arthur C. Graesser, Danielle S. McNamara, and Jonna M. Kulikowich. 2011. Coh-metrix: Providing multilevel analyses of text characteristics. *Educational Researcher*, 40(5):223–234.

Stephen Grossberg. 1988. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, 1(1):17–61.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of Advances in Neural Information Processing Systems*, pages 1693–1701.

Sepp Hochreiter. 1991. Untersuchungen zu dynamischen neuronalen netzen. *Diploma, Technische Universität München*, 91:1.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Ikasbil. 2018. Resources. `http://www.ikasbil.eus`.

Lilian Kandel and Abraham Moles. 1958. Application de l'indice de flesch à la langue française. *Cahiers Etudes de Radio-Télévision*, 19:253–274.

Tapas Kanungo and David Orr. 2009. Predicting the readability of short Web summaries. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 202–211.

Nikolay Karpov, Julia Baranova, and Fedor Vitugin. 2014. Single-sentence readability prediction in russian. In *Proceedings of the International Conference on Analysis of Images, Social Networks and Texts*, pages 91–100.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lexile. 2016. How is the lexile measure of a text determined? Available at: `https://lexile.desk.com/customer/en/portal/articles/508829-how-is-the-lexile-measure-of-a-text-determined-`.

Pietro Lucisano and Maria Emanuela Piemontese. 1988. Gulpease: una formula per la predizione della difficoltà dei testi in lingua italiana. *Scuola e città*, 3(31):110–124.

Ion Madrazo Azpiazu. 2017. Towards multilingual readability assessment. Master's thesis, Boise State University, Boise, Idaho.

Ion Madrazo Azpiazu, Nevena Dragovic, Oghenemaro Anuyah, and Maria Soledad Pera. 2018. Looking for the movie Seven or sven from the movie Frozen?: A multi-perspective strategy for recommending queries for children. In *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval (ACM CHIIR)*, pages 92–101.

Bangalore S. Manjunath and Wei-Ying Ma. 1996. Texture features for browsing and retrieval of image data. *Institute of Electrical and Electronics Engineers Transactions on Pattern Analysis and machine Intelligence*, 18(8):837–842.

Ion Madrazo. 2014. Testuen irakurgarritasuna neurtzeko sailkatzaile automatikoa [an automatic classifier of text legibility]. Master's thesis, University of the Basque Country (UPV/EHU). Advisor: Montse Maritxalar Anglada.

Mohsen Mesgar and Michael Strube. 2015. Graph-based coherence modeling for assessing readability. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 309–318.

Mohsen Mesgar and Michael Strube. 2016. Lexical coherence graph modeling using word embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1414–1423.

Mohsen Mesgar and Michael Strube. 2018. A neural local coherence model for text quality assessment. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4328–4339.

Gustavo H. Paetzold and Lucia Specia. 2016. Unsupervised lexical simplification for non-native speakers. In *Association for the Advancement of Artificial Intelligence*, pages 3761–3767.

Maria Soledad Pera and Yiu-Kai Ng. 2014. Automating readers' advisory to make book recommendations for k-12 readers. In *Association of Computer Machinery Conference on Recommender Systems*, pages 9–16.

Sarah E. Schwarm and Mari Ostendorf. 2005. Reading level assessment using support vector machines and statistical language models. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 523–530.

Seth Spaulding. 1956. A Spanish readability formula. *The Modern Language Journal*, 40(8):433–441.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for Twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1555–1565.

Jin Wang, Liang-Chih Yu, K. Robert Lai, and Xuejie Zhang. 2016. Dimensional sentiment analysis using a regional CNN-LSTM model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 225–230.

Wikimedia. 2018. Simplification guidelines for simple wikipedia. https://simple.wikipedia.org/wiki/Wikipedia:Simple_English_Wikipedia.

Wizenoze. 2018. Resources. http://www.wizenoze.com.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015a. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015b. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning*, pages 2048–2057.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.