Statistical Measures for Readability Assessment

Mohammed Attia
Google LLC,
US
attia@google.com

Younes Samih
IBM Research,
UAE
younes.samih@ibm.com

Yo Ehara
Tokyo Gakugei University,
Japan
ehara@u-gakugei.ac.jp

Abstract

Neural models and deep learning techniques have predominantly been used in many tasks of natural language processing (NLP), including automatic readability assessment (ARA). They apply deep transfer learning and enjoy high accuracy. However, most of the models still cannot leverage long dependence such as inter-sentential topic-level or document-level information because of their structure and computational cost. Moreover, neural models usually have low interpretability. In this paper, we propose a generalization of passage-level, corpus-level, document-level and topic-level features. In our experiments, we show the effectiveness of "Statistical Lexical Spread (SLS)" features when combined with IDF (inverse document frequency) and TF-IDF (term frequency–inverse document frequency), which adds a topological perspective (inter-document) to readability to complement the typological approaches (intra-document) used in traditional readability formulas. Interestingly, simply adding these features in BERT models outperformed state-of-the-art systems trained on a large number of hand-crafted features derived from heavy linguistic processing. In analysis, we show that SLS is also easy-to-interpret because SLS computes lexical features, which appear explicitly in texts, compared to parameters in neural models.

1 Introduction

A large number of readability formulas (also called shallow readability indicators) have been developed since the 1940's, but most of them use superficial intra-sentential information (e.g., average sentence length and average character length) without using inter-sentential information such as document-level, corpus-level and topic-level statistics.

To address this issue, we introduce Statistical Lexical Spread (SLS), and combine it with features derived from IDF and TF-IDF to train neural and non-neural models on automatic readability assessment (ARA). This set of data-driven features can be extracted from any corpus, preferably where documents are categorized into topics. In this project we utilize Wikipedia where articles, by design, are grouped into categories.

We use these features to augment a BERT-Based classifier on some benchmark data sets to determine if any significant improvement can be gained from these features or they are already learned by BERT embeddings. For this purpose we develop a 'single-shot' model where BERT is fine-tuned on the text alone or text combined with the numerical values of our features, and a 'hybrid' model, where the BERT pipeline is augmented with the predictions from a non-neural classifier. Interestingly, the 'hybrid' mode shows remarkable improvement on the results of the 'single-shot' mode, and overall our models outperform (or compete with) state-of-the-art methods that rely on heavy linguistic processing.

To test the generalizability and crosslinguality of our methods, we evaluate our models on English, Spanish, and Catalan. The advantage of our approach is that no sophisticated NLP processing tools or resources are needed, apart from an optional lemmatizer, which makes it suitable for low-resourced languages. In case a lemmatizer does not exist, SLS+TF-IDF can still be narrowed down to statistics on the surface forms with even better performance on some data sets, while on others yielding only 1.43% absolute below the highest scores.

2 Related Work

With the recent advancement of machine learning (ML), researchers started to apply it to ARA usually modeling it as a classification task. Early studies introducing ML to ARA developed hand-crafted features extracted mostly from the linguistic analysis of texts. For example, Schwarm and Osten-

Proceedings of the Joint 3rd NLP4DH and 8th IWCLUL, pages 153–161

dorf (2005) introduced four syntactic features (average parse tree height, and average number of noun phrases, verb phrases, and SBARs) to train an SVM classifier. Pitler and Nenkova (2008) enriched that with features indicating lexical cohesion (e.g. number of pronouns in a sentence, and word overlap between sentences), and discourse connectivity (e.g. whether connectives between sentences are implicit or explicit). Over the years, the number of linguistic features kept growing reaching 155 (Vajjala and Lučić, 2018) and 255 (Lee et al., 2021).

Although the focus on generating sophisticated linguistic features might help advance the state of the art for English, it does not generalize well, due to the fact that many languages have limited NLP tools and resources. For example, (Imperial, 2021) used 155 linguistic features for English and only 54 for Filipino due to this limitation. This is why we introduce frequency-based features with minimal NLP tooling requirements (only a lemmatizer) that scales well across languages. We also show that even with the lack of a lemmatizer, the non-morphology based features can still deliver a comparable performance.

There have been a few attempts to depart from linguistic features for ARA, particularly using the help of language models. For example, Collins-Thompson and Callan (2004) developed 12 language models matching the 12 American grade levels. Their language models are unigrams and assume that the probability of a token given the grade level is independent of the surrounding tokens. Cha et al. (2017) used Brown clustering which aims to maximize the mutual information of word bigrams. Language models, however, focus on corpus-level information, and they do not have a mechanism to account for passage-level, document-level or topic-level information, which TF-IDF and readability formulas, for example, prove to be more suited for.

3 Data Collection and Sampling

As is the case with document indexing in Information Retrieval and Text Mining, for the construction of data-driven features, we need to compute weights for each word to quantify the degree of its familiarity. Instead of using a set of web pages, we use Wikipedia articles for our indexing purposes. There are three primary advantages of Wikipedia for our approach to ARA: first, it is available in many languages, second, it covers a broad variety of topics, and third, most articles are associated with categories, which allows us to cluster articles into their related topics. However, the disadvantage is that Wikipedia articles are edited and reviewed to be of a high quality, and therefore lack the noise and variance common in many other natural text types.

Data point	Count
total titles	6,334,131
total categories	1,347,602
total word count	2,363,334,969
titles with categories	4,049,500
singleton categories	310,997
categories $\in 80\%$ of titles	1,161
* titles $\in 80\%$ of word count	1,812,671
* categories $\in 80\%$ of '* titles'	1,131

Table 1: Topography of titles and categories in the English Wikipedia. *: included in the final selection.

Due to the large number of titles in the English Wikipedia, and the fact that many articles are seed articles, i.e. without any substantial content, we sub-sample the data following Pareto's Principle which states that 80% of consequences come from 20% of the causes. For a total number of 6.3m articles we found that 28.62% of them cover 80% of the word count. By contrast, there are 1.3m categories, and we found that 0.09% of them cover 80% of the titles. The reason that category selection seems to go off the bounds for Pareto's Principle is that categories are very liberally used in Wikipedia. For example, 23.08% of the categories are singletons, i.e. representing only one article. Statistics for the English Wikipedia data dump of September 2^{nd}, 2021 are shown in Table 1. The same sub-sampling strategy is used for the other two languages tested in this project, i.e. Spanish and Catalan.

4 Feature Design and Selection

Understanding a document is dependent on the reader's level of familiarity with the underlying knowledge base (or the topic), which accounts for the connections in the mental map (Liu and Yuizono, 2020) of the reader and controls the flow of information for updating these connections. This underlying knowledge base indicates the presence or absence of the shared world knowledge between the writer and the reader. Approximating this underlying knowledge map can be obtained by analyzing

the connection between words within a document and across a reasonably large collection of documents.

Topic modeling for ARA has been discussed in a number of papers. For example, Qumsiyeh and Ng (2011) developed a system called ReadAid that used Latent Dirichlet Allocation (LDA), an unsupervised learning algorithm, to determine the ranked probability of topics covered in a document based on the distribution of words in the document, or more precisely the probability of a word given a topic $P(w|t)$, and the probability of a topic given a document $P(t|d)$. Their model was trained on 53 subject areas extracted from the English Curriculum and College Board[1] and a set of 100 documents randomly selected for each subject area from DMOZ[2]. Lee et al. (2021) expanded this approach by training LDA models on four variations of 50, 100, 150, and 200 topics, and analyzed the output for semantic richness, clarity, and noise, with the purpose of understanding how the topics are distributed, not just what they are.

In this work we use SLS+TF-IDF to model topics using top-frequency categories already curated in the Wikipedia corpus, with the intuition that specialized words will occur in fewer categories than common ones. The categories selected in our analysis are based on a ratio of titles and word counts as explained in Section 3, and the number is 1,131 for English, 1,536 for Spanish, and 1,516 for Catalan.

The features used in this project are divided into four groups: Statistical Lexical Spread (SLS), TF-IDF features, document-based counts, and traditional readability formulas (RF). Details are explained in the following sub-sections.

4.1 Statistical Lexical Spread (SLS)

The main intuition for SLS is that easy words occur more often and in more contexts, spanning more articles and more topics, than difficult words. Even if a word is long and multi-syllabic, such as 'television', if it occurs more often, it will be considered more readable than less frequent words, even if they are short and monosyllabic, such as 'deuce'. Another intuition is that words which show a high morphological variability, such as 'play, plays, played, playing', are generally easier to read than rigid and uninflected words, such as 'timid'. The advantage in SLS is that frequency statistics are gathered at the corpus level, document level and topic level.

For the three features of 'unknown_word', 'uninflected_word' and 'below_mean_count' we just take the ratio (count of positive tokens divided by the total number of tokens in a document). For the other seven features we take the log of the average according to Equation 1, where t is a term which can be a lemma or a form, $f(t)$ is the function that retrieves the frequency, spread or variability value, and $l(d)$ is the length of the document. Features with the suffix '$\_freq$' are for corpus-level statistics, '$\_article\_spread$' for document-level statistics, and '$\_category\_spread$' for topic-level statistics. For English lemmatization, we use NLTK (Bird et al., 2009), and for Spanish and Catalan, we use spaCy (Honnibal and Montani, 2017).

$$\log\left(\frac{\sum_{t=1}^{l(d)} f(t)}{l(d)}\right) \quad (1)$$

Non-Morphology Features:

1. form_freq: form frequency, or how many times a form occurred in the entire corpus.

2. form_article_spread: in how many articles a form appeared, regardless of total frequency.

3. form_category_spread: in how many categories a form appeared.

4. unknown_word: words that do not occur in the corpus or have a frequency below a certain threshold, which is set in our experiment to 10.

5. below_mean_count: words that have a frequency below the mean frequency of the word list consumed (excluding unknown words above). This happens to be 1441.79 in the English Wikipedia sample.

Morphology-based Features:

6. lemma_freq: lemma frequency, or how many times a lemma occurred in the entire corpus.

7. lemma_article_spread: in how many articles a lemma appeared, regardless of the total frequency.

8. lemma_category_spread: in how many categories a lemma appeared.

9. morph_variability: for each lemma, how many different forms are represented by the given lemma. This is an indication of morphological richness.

10. uninflected_word: words that do not have any morphological inflection in the corpus.

The use of the log in the calculations is meant as a normalization step to dampen the effect of exploding numbers when the numerator is much greater

[1] www.collegeboard.org

[2] www.dmoz-odp.org

than the denominator and the variance between different outputs cannot fit in a scale.

4.2 TF-IDF Features

TF-IDF has been used in the readability literature in two different ways.

1. Using the TF-IDF for all tokens in a given document as a vector (Chen et al., 2011).
2. Using the mean of TF-IDF of all tokens a document (De Clercq et al., 2014).

TF-IDF is powerful in collecting statistics on term distribution and weight across a collection of documents. In our research, we use the mean of TF-IDF and the mean of the IDF for forms and lemmas in a document. This gives us 4 features. We further apply it to articles, and categories as documents (for topic modeling). This expands the number of features to 8, and this allows us to utilize the power of the TF-IDF orthogonally at the document level and the topic level. Equations 2, 3, and 4, show how the calculations are conducted, where t is the term which can materialize as a form or a lemma, d is a document, D is a collection of documents (or categories), l is the length function, c is the counting function, e.g. $c(t, d)$ is the count of term repetitions in a given document, and uc is a unique counting function, i.e. if a term occurs in a document one or more times, it will be reduced to one, otherwise zero.

$$TF_{(t,d)} = \frac{c(t,d)}{l(d)} \quad (2)$$

$$IDF_{(t,D)} = \log\left(\frac{l(D)}{uc(t,D)}\right) \quad (3)$$

$$TF\text{-}IDF_{(t,d,D)} = TF_{(t,d)} \times IDF_{(t,D)} \quad (4)$$

Here we list the features derived from TF-IDF also divided into whether they are dependent/non-dependent on morphological analysis (lemmatization).

Non-Morphology Features:

1. form_article_idf: average IDF where t is a word form and D is a collection of articles.

2. form_category_idf: average IDF where t is a word form and D is a collection of categories.

3. form_article_tf-idf: average TF-IDF where t is a word form and D is a collection of articles.

4. form_category_tf-idf: average TF-IDF where t is a word form and D is a collection of categories.

Morphology-based Features:

5. lemma_article_idf: average IDF where t is a word lemma and D is a collection of articles.

6. lemma_category_idf: average IDF where t is a word lemma and D is a collection of categories.

7. lemma_article_tf-idf: average TF-IDF where t is a word lemma and D is a collection of articles.

8. lemma_category_tf-idf: average TF-IDF where t is a word lemma and D is a collection of categories.

4.3 Document-Based Features

We need to account for passage-level information, such as a word repetition, word count and the type of lexicon used. These features are computed locally by counting words in a given document, or matching them against predefined lists.

1. word_count: word count in the current document, taken as a ratio against the maximum word count found in a document set.

2. word_rep: in a given document, how many times a word is repeated. This is then averaged against total words in a document

3. basic_vocab: how many words are found in a list of basic vocabulary. The source of the word list is Simple Wikipedia list of 1000 basic words. This is taken as a ratio against total words in a document.

4.4 Readability Formulas (RF):

Readability Formulas (RF) are known for their efficiency at capturing passage-level information. There are a few python implementations of these formulas. In this project, we chose the implementation in TextStat.[3] Here is a list of the formulas used:

1. Flesch Reading Ease, (Kincaid et al., 1975).
2. Flesch-Kincaid Grade, (Kincaid et al., 1975).
3. SMOG Index, (Mc Laughlin, 1969).
4. Coleman-Liau Index, (Coleman and Liau, 1975).
5. Automated Readability Index, (Smith and Senter, 1967).
6. Dale-Chall Readability Score, (Dale and Chall, 1948)
7. Linsear Write Formula[4].
8. Gunning-Fog Index, (Gunning et al., 1952).
9. Text Standard, based on consensus among a number of tests.

[3] https://github.com/textstat/textstat

[4] https://en.wikipedia.org/wiki/Linsear_Write

10. Fernandez-Huerta, (Fernández-Huerta, 1959).
11. Szigriszt-Pazos, (Szigriszt Pazos, 1992).
12. Gutierrez Polini, (Gutiérrez de Polini, 1972).
13. Crawford, (Crawford, 1985).
14. Gulpease Index[5].
15. Osman, (El-Haj and Rayson, 2016).
16. Difficult Words, (Dale and Chall, 1948).

4.5 Feature Subsets

For some ML algorithms, the high dimensionality of features can be problematic. Therefore, we use XGBoost to determine the important features based on training on the English Viki-Wiki data set. We select the overlap between gain and coverage, and here are the subsets selected.
Selected_8 = (4.1): 4, (4.1): 2, (4.4): 8, (4.4): 6, (4.2): 4, (4.2): 1, (4.4): 7, (4.3): 1
Selected_6 = first 6 in selected_8.
SLS_8 = (4.1): 6, (4.1): 2, (4.1): 1, (4.1): 7, (4.1): 9, (4.1): 4, (4.1): 3, (4.1): 8
SLS_6 = first 6 in sls_8.
RF_8 = (4.4): 8, (4.4): 6, (4.4): 7, (4.4): 11, (4.4): 2, (4.4): 3, (4.4): 13, (4.4): 15
RF_6 = first 6 in rf_8.

5 Correlation with Readability Formulas

We conducted a comparison between our statistical measures and the traditional readability formulas to see to what degree they are aligned on their predictions. The data set used in this experiment is the Simple Wikipedia, with a total number of instances of 142,759. We used a split of 66% for training and 34% for testing. We applied the decision tree Random Forest algorithm, and the results are shown in Figure 1.

Generally, there seems to be a strong correlation between our SLS and dale_chall_readability_score, while document-based and TF-IDF have the highest correlation with 'difficult_words'. We notice that some non-English specific indicators, such as 'osman', 'gulpease_index' and 'gutierrez_polini' have higher correlation with our criteria than some English-specific ones, such as 'gunning_fog' and 'smog_index'. This is why we decided to use all 15 formulas in subsequent experiments.

It's also interesting to consider the correlation coefficient among the different traditional readability formulas. Many pairs of formulas have high correlation, whether negative or positive, which means that they are looking at the same or similar pieces of information, while many other pairs have a correlation between -0.5 and 0.5 which indicates low correlation, meaning they are looking at different pieces of information, or interpreting the same pieces of information differently.

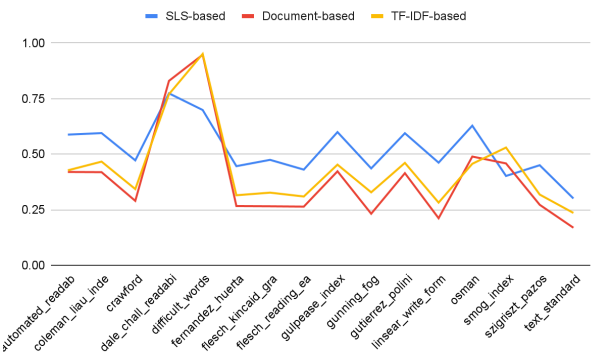

Figure 1: Correlation between new and traditional readability formulas

6 Testing on Benchmark Test Sets

We use our features, along with the readability scores from the traditional readability formulas and build ML models and apply them to two benchmark data sets: a monolingual one, OneStopEnglish, and a multilingual one, VikiWiki.

6.1 OneStopEnglish (OSE)

OneStopEnglish (Vajjala and Lučić, 2018) is a collection of articles obtained from the Guardian newspaper and adapted by teachers for three levels of learners (elementary, intermediate, and advanced). The data set contains 564 instances (189 elementary, 189 intermediate, and 186 advanced.

6.1.1 Classification with Non-neural Classifiers

Table 2 shows the results of the experiments with 10-fold cross validation using a number of non-neural ML classifiers. Our best models give an accuracy of 80.15% using the 'selected_8_no_morph' features in an SVM classifier. This outperforms the results in (Vajjala and Lučić, 2018) which was 78.13% using 155 linguistic features, and the results in (Lee et al., 2021) which was 77.8% using 255 handcrafted features.

In our initial experiments, we noticed that the results for train-test splits can vary dramatically by the split size, while n-fold cross validation gives

[5] https://it.wikipedia.org/wiki/Indice_Gulpease

Feature sets	LR	SVM	XGB	RF
SLS	43.24	38.97	35.42	31.54
Doc-based	65.58	69.51	67.72	66.67
TF-IDF	66.11	68.45	64.71	63.84
RF	69.85	74.82	75.19	74.12
selected_8	72.16	79.26	77.31	78.38
selected_8 _no_morph	72.15	**80.15**	77.14	77.66
selected_6	69.14	78.02	78.37	77.48
SLS_8	42.52	39.69	34.88	33.85
SLS_6	39.70	37.38	35.61	32.92
RF_8	59.58	60.65	57.27	58.88
RF_6	59.22	60.47	55.85	57.79
All features	73.93	77.48	75.52	77.66

Table 2: ML Classification results on OSE. LR = Logistic Regression, XGB = XGBoost, RF = RandomForest

more reliable results, particularly for smaller data sets.

6.1.2 Classification with BERT Fine-Tuning

Since its inception, BERT (Devlin et al., 2018) has shown strong performance on many NLP benchmark data sets. There were a number of attempts to apply it to ARA, including that of Martinc et al. (2021), who reported an accuracy of 67.38% training on text alone without additional features. Their best result on the OSE data set was 78.72% using HAN (Hierarchical attention networks).

Combining BERT embeddings with additional features has been explored in a number of papers and most of them used the fused features in a non-neural classifier (Deutsch et al., 2020). For example, (Imperial, 2021) extracted BERT embeddings, concatenated them with 155 linguistic features, making a total of 923 dimensions, and fed that into a number of ML classifiers. Imperial (2021)'s best result was an F1 score of 73.2% using logistic regression.

In a more recent paper, (Lee et al., 2021) reported 80.1% mean accuracy on five-folds on OSE using BERT without handcrafted features. They further managed to increase the accuracy to 98.2% using a hybrid model, where they took the predictions of BERT fine-tuning, along with 255 handcrafted features, and fed them to a non-neural classifier.

In this paper, we experiment with BERT in two modes: 'Single-Shot Mode' and 'Hybrid Mode', as explained below. For English we use the model 'bert-base-uncased' for English and 'bert-base-multilingual-uncased' for the other languages.

Parameters	values
Epochs	16
Learning rate	1e-5
max token length	200
batch size	32
optimizer	AdamW
number of folds	10

Table 3: BERT classification setup.

Single-Shot Mode: Here we combine our features with the text embeddings following Chris McCormick article on "Combining Categorical and Numerical Features with Text in BERT".[6] We use the model 'transformers.BertForSequenceClassification' with the parameters listed in Table 3.

We tried two ways of appending the numerical values of features to the text. The first was to include the numerical values separators, and the second was to concatenate the feature name along with the numerical value. We found that the second method worked best, and this is what is reported in this paper.

1. `f'{value} [SEP]. '`
2. `f'{feature}: {value} [SEP]. '`

Hybrid Mode: Similar to Lee et al. (2021), we also build a hybrid model, but instead of using BERT predictions in a non-neural model, we use the predictions of a non-neural model and feed them to the BERT fine-tuning along with the feature sets and the text embeddings. We first train SVM on 'selected_8_no_morph', take the predictions for each fold (so that there is no chance for over-fitting), and combine them together as an additional feature in BERT fine-tuning.

Results for both the single-shot and the hybrid model are shown in Table 4. All experiments are conducted with 10-fold cross-validation. Our baseline is BERT fine-tuned on text embeddings only without any features, which is 86.16% for the single-shot model. This is significantly greater than the 80.1% reported by Lee et al. (2021). The best result for the single-shot mode was 96.64% when BERT embedding is concatenated with the

[6] https://mccormickml.com/2021/06/29/combining-categorical-numerical-features-with-bert/

Features combined	Single-Shot	Hybrid
No features used	86.16	93.42
All features used	39.54	77.84
SLS	29.09	77.84
Doc-based	75.55	91.84
TF-IDF	**96.64**	**98.23**
TF-IDF_no_morph	88.51	96.80
RF	70.94	82.63
Selected_8	77.83	97.52
Selected_6	86.31	95.56
SLS_8	87.42	98.05
SLS_6	87.01	95.38
RF_8	81.72	91.12
RF_6	79.61	84.56

Table 4: BERT fine-tuning results on OSE.

eight features of TF-IDF. For the hybrid mode, our best result is 98.23%. We notice that the hybrid mode gives a significant boost to the performance on most of the features used.

It must be noted that (Lee et al., 2021) managed to obtain 96.5%, and 96.8% accuracy in single-shot mode using RoBERTa and BART respectively, and 99.0%, and 97.1% in a hybrid mode.

6.2 Viki-Wiki

VikiWiki is a multilingual readability data set of Vikidia articles and their Wikipedia counterparts (Madrazo Azpiazu and Pera, 2020). Vikidia[7], like Wikipedia, is an encyclopedic website providing information on various topics in English and a number of other languages, but the main goal is to make the content simple and easy to read. The number of instances in the dataset is 864 for English, 831 for Spanish, and 855 for Catalan. The best results reported by Madrazo Azpiazu and Pera (2020) (in terms of accuracy for 10-fold cross-validation) was 96% for English, 87% for Spanish and 96% for Catalan. In their work, Madrazo Azpiazu and Pera (2020) used different sets of features including shallow, morphological, syntactic, and semantic features.

6.2.1 Classification with non-neural Classifiers

Table 5 shows the results of our system trained on a combination of features. Our results are comparable to those of (Madrazo Azpiazu and Pera, 2020) for English, Spanish and Catalan. We found that

[7] https://en.vikidia.org

Features sets	en	es	ca
SLS	90.62	82.68	94.38
Doc-based	93.98	84.72	95.56
TF-IDF	92.48	82.32	94.15
RF	95.60	84.61	94.62
Selected_8	95.37	86.65	95.21
Selected_6	**95.95**	86.05	95.09
Selected_6 _no_morph	95.26	84.49	94.63
SLS_8	89.12	81.72	93.57
SLS_6	89.93	81.95	92.05
RF_8	90.74	83.88	93.80
RF_6	90.16	84.37	94.04
All_features	95.37	86.77	**95.79**
All_features _no_morph	95.49	**87.49**	**95.79**

Table 5: XGBoost Classification results on Viki-Wiki.

XGBoost gives the best performance compared to other ML algorithms (and this is why we report XGBoost results only here). Again all experiments are conducted with 10-fold cross-validation.

6.2.2 Classification with BERT Fine-tuning

Following the same approach above with OSE in concatenating numerical values to text in BERT embedding in a **single-shot mode** (Section 6.1.2), we conducted 10-fold cross-validation experiments for English, Spanish and Catalan. The results are shown in Table 6. In most cases the performance converges to 100%. We are not entirely sure about the reason, but it can be due to the fact that BERT is already trained on Wikipedia data, or the task is too easy as the language in the two data sets is clearly distinct. The best published results in the literature is 96% for English and Catalan, which is already high.

Conclusion

For ARA, hand-crafted features derived from heavy linguistic processing do not transfer well across languages, as it becomes harder to find reliable processing tools for low-resourced languages. Our system, by contrast, achieves better or comparable results using only 38 features that capture passage-level, corpus-level, document-level, and topic-level information, and can be computed statistically from any corpus in any language with a light-weight morphological processing tool. We show that even with the absence of a lemmatizer, non-morphological

Features sets	en	es	ca
SLS	83.10	80.10	92.84
Doc-based	99.13	100.00	99.88
TF-IDF	100.00	99.88	99.87
RF	99.89	99.71	99.87
Selected_8	100.00	100.00	100.00
Selected_6	100.00	100.00	99.88
SLS_8	100.00	100.00	99.87
SLS_6	100.00	100.00	100.00
RF_8	100.00	100.00	100.00
RF_6	100.00	100.00	100.00
No features	99.89	100.00	99.86
All features	94.42	79.61	90.69

Table 6: BERT Classification results on Viki-Wiki.

features can still yield comparable results. We also show how topic modeling for ARA can be achieved through treating categories as documents in computing features such as IDF and TF-IDF.

Limitations

SLS+TF-IDF provides information on word difficulty and topical specificity drawn from actual language use. One presumed shortcoming of the proposed approach is that it focuses on the lexical statistical behavior and ignores semantic, syntactic and discourse features. Due to the utilization of a lemmatizer, the system can distinguish between 'flag' as a noun and a verb, but it will not be able to distinguish between 'lead' as a metal and or a leash.

Another limitation of the results is that the use of BERT and neural net, by nature, gives different results each run. Although we use 10-fold cross-validation and a relatively higher number of epochs to narrow down the effect of this variability, it is still possible to get slightly different results for each run.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Miriam Cha, Youngjune Gwon, and HT Kung. 2017. Language modeling by clustering with word embeddings for text readability assessment. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2003–2006.

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

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

Kevyn Collins-Thompson and James P Callan. 2004. A language modeling approach to predicting reading difficulty. In *Proceedings of the human language technology conference of the North American chapter of the association for computational linguistics: HLT-NAACL 2004*, pages 193–200.

A Crawford. 1985. Fórmula y gráfico para determinar la comprensibilidad de textos del nivel primario en castellano. *Lectura Y Vida*, 4:18–24.

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.

Orphée De Clercq, Véronique Hoste, Bart Desmet, Philip Van Oosten, Martine De Cock, and Lieve Macken. 2014. Using the crowd for readability prediction. *Natural Language Engineering*, 20(3):293–325.

Tovly Deutsch, Masoud Jasbi, and Stuart Shieber. 2020. Linguistic features for readability assessment. *arXiv preprint arXiv:2006.00377*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mahmoud El-Haj and Paul Edward Rayson. 2016. Osman: A novel arabic readability metric.

J Fernández-Huerta. 1959. Medidas sencillas de lecturabilidad [simple readability measures]. *Consigna*, 214:29–32.

Robert Gunning et al. 1952. Technique of clear writing.

L.E. Gutiérrez de Polini. 1972. Investigación sobre lectura en venezuela. *las Primeras Jornadas de Educación Primaria*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Joseph Marvin Imperial. 2021. Bert embeddings for automatic readability assessment. *arXiv preprint arXiv:2106.07935*.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for

navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Bruce W Lee, Yoo Sung Jang, and Jason Hyung-Jong Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. *arXiv preprint arXiv:2109.12258*.

Ting Liu and Takaya Yuizono. 2020. Mind mapping training's effects on reading ability: Detection based on eye tracking sensors. In *Sensors (Basel, Switzerland) vol. 20,16 4422. 7 Aug. 2020, doi:10.3390/s20164422*.

Ion Madrazo Azpiazu and Maria Soledad Pera. 2020. Is cross-lingual readability assessment possible? *Journal of the Association for Information Science and Technology*, 71(6):644–656.

Matej Martinc, Senja Pollak, and Marko Robnik-Šikonja. 2021. Supervised and unsupervised neural approaches to text readability. *Computational Linguistics*, 47(1):141–179.

G Harry Mc Laughlin. 1969. Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 186–195.

Rani Qumsiyeh and Yiu-Kai Ng. 2011. Readaid: a robust and fully-automated readability assessment tool. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, pages 539–546. IEEE.

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 of the Association for Computational Linguistics (ACL'05)*, pages 523–530.

Edgar A Smith and RJ Senter. 1967. *Automated readability index*, volume 66. Aerospace Medical Research Laboratories.

Francisco Szigriszt Pazos. 1992. Sistemas predictivos de legilibilidad del mensaje escrito: fórmula de perspicuidad.

Sowmya Vajjala and Ivana Lučić. 2018. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 297–304, New Orleans, Louisiana. Association for Computational Linguistics.