IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 29, 2021, accepted April 7, 2021, date of publication April 12, 2021, date of current version April 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3072697

Lexical Simplification System to Improve Web Accessibility

RODRIGO ALARCON, LOURDES MORENO, AND PALOMA MARTÍNEZ
Computer Science and Engineering Department, Universidad Carlos III de Madrid, 28911 Madrid, Spain

Corresponding authors: Rodrigo Alarcon (ralarcon@inf.uc3m.es), Lourdes Moreno (lmoreno@inf.uc3m.es), and Paloma Martínez (pmf@inf.uc3m.es)

This work was supported in part by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with UC3M in the line of Excellence of University Professors under Grant EPUC3M17, in part by the context of the V PRICIT (Regional Programme of Research and Technological Innovation), and in part by the Accessible Technologies Award-INDRA Technologies and the Fundación Universia (www.tecnologiasaccesibles.com).

ABSTRACT People with intellectual, language and learning disabilities face accessibility barriers when reading texts with complex words. Following accessibility guidelines, complex words can be identified, and easy synonyms and definitions can be provided for them as reading aids. To offer support to these reading aids, a lexical simplification system for Spanish has been developed and is presented in this article. The system covers the complex word identification (CWI) task and offers replacement candidates with the substitute generation and selection (SG/SS) task. These tasks have followed machine learning techniques and contextual embeddings using Easy Reading and Plain Language resources, such as dictionaries and corpora. Additionally, due to the polysemy present in the language, the system provides definitions for complex words, which are disambiguated by a rule-based method supported by a state-of-the-art embedding resource. This system is integrated into a web system that provides an easy way to improve the readability and comprehension of Spanish texts. The results obtained are satisfactory; in the CWI task, better results were obtained than with other systems that used the same dataset. The SG/SS task results are comparable to similar works in the English language and provide a solid starting point to improve this task for the Spanish language. Finally, the results of the disambiguation process evaluation were good when evaluated by a linguistic expert. These findings represent an additional advancement in the lexical simplification of texts in Spanish and in a generic domain using easy-to-read resources, among others, to provide systematic support to compliance with accessibility guidelines.

INDEX TERMS Accessibility, contextualized word embeddings, lexical simplification, natural language processing, Spanish language, readability.

I. INTRODUCTION

The readability and understandability of texts containing long sentences, unusual words and complex linguistic structures can result in cognitive accessibility barriers for individuals with intellectual disabilities. However, the benefits of offering simplified text content will be enjoyed by individuals with intellectual and learning disabilities and deaf and deaf-blind individuals, the elderly, the illiterate, and immigrants whose native language is different, among others.

According to the OECD survey of adult skills [1], nearly 19 percent of adults in Europe have poor reading skills. Thus, a significant portion of the population cannot read documents containing large amounts of information; thus, it needs to be simplified [2]. According to the Programme for International Student Assessment (PISA) report, most adults in Spain have difficulties understanding dense texts [3]. In addition, 1.7% of the population is functionally illiterate, and there are 277,472 people with some type of intellectual disability in Spain.

The existing directives provide accessibility guidelines regarding how to make content more accessible for individuals with intellectual and learning disabilities [5], [7]–[10]. These include criteria that involve offering resources that provide text simplification, which is difficult to fulfil. Few tools exist that provide systematic support for simplification processes. Usually, the websites that offer simplified versions of their main sites are manually created. Manual simplification

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna Dulizia.

of written documents is quite expensive, mostly because the information is continually being produced. As part of the solution, text simplification methods, which are found in the natural language processing (NLP) field, provide systematic support to promote compliance with these cognitive accessibility guidelines. This is the motivation behind this work.

After an analysis of language accessibility guidelines, this work presents a system to support the lexical simplification processes applied to text content in the Spanish language to improve its readability and understandability.

This system follows a pipeline; the first step identifies complex words following a machine learning approach using Easy-to-Read features. As a next step, simpler terms for each complex word are offered through generation and selection of substitute processes. The generation process uses various Spanish language linguistic databases. For substitute selection, from a list of synonyms extracted in the previous step, the most suitable synonym is selected according to its simplicity and the context using word embedding methods. In addition, to complement the lexical simplification processes, the system provides definitions for the detected complex words, and due to the number of polysemous words present in the language, a disambiguation procedure to determine the meaning of the words is proposed using a contextual method supported by a BERT model. The results are satisfactory compared to relative work. The findings are promising because (1) we propose a new combination of features to discern between a complex word and simple word; (2) we offer contextualized, less complex replacements to the complex word by using embedding and linguistic resources; and (3) we provide a new procedure to generate a context-aware simpler definition for a target word.

This contribution is integrated into the EASIER web system, which offers additional aid complements to improve comprehension and readability for the final user.

The remainder of this paper is organized as follows. In Section II, we explore the accessibility directives that led us to the objectives of this work. In Section III, we review the work related to the text simplification process. Section IV describes the simplification approach. Sections V, VI, VII and VIII provide the procedures and evaluate the complex word identification, substitute generation/selection and word-sense-disambiguation modules. Section IX includes a discussion of this work. Section X presents the EASIER system as a proof of concept. Finally, Section XI offers conclusions and suggestions for future work.

II. ACCESSIBILITY REQUIREMENTS

Some directives provide guidelines for making text content more accessible for individuals with intellectual, language, and learning disabilities, which are introduced below:

The Web Content Accessibility Guidelines (WCAG) [5], which is part of the W3C's WAI (Web Accessibility Initiative) and is the benchmark standard followed by most regulatory frameworks [6], is one such an initiative. The WCAG includes specific guidelines that help make web content accessible for individuals with intellectual and learning disabilities. Another important initiative to be considered is the ''Cognitive and Learning Disabilities Accessibility Task Force (W3C-COGA TF)'' [12]. Another important initiative is the easy-to-read guidelines. These guidelines have been disseminated thanks to the work of the International Federation of Library Associations and Institutions (IFLA), which published the document titled ''Directives for Easy-to-Read Materials'' [7]. Additionally, Inclusion Europe (Formerly the International League of Societies for Persons with Mental Handicap (ILSMH)) published the document titled ''Make it Simple: European Guidelines for the Production of Easy-to-Read information'' [8]. The standard regarding easy-to-read content in Spanish was considered in this work [11]. In addition to the Easy-to-Read initiative, the Plain Language initiative is geared towards the general public [9], [10]. It was created to promote simple language in all electronic governmental content and information and provide improved customer service to all citizens. Currently, providing accessible information in e-administration, e-health and other services is a priority in many governments. For this reason, they are developing guides and adapting many of their public communications (plainlanguageeurope.com/en).

An analysis of these guidelines was performed and is shown in Table 1. Although some differences are found among these initiatives, certain overlap can be observed.

Note that using a simple lexicon is an element that is repeated in all the guidelines, as shown in Table 1. Individuals with language impairments often have a reduced vocabulary, and learning new terms is a slow and challenging process. WCAG 2.1 and COGA documentation have been considered to provide solutions and comply with this guideline. The WCAG Success Criterion 3.1.3 (Unusual Words) indicates that a mechanism must be made available to identify specific definitions of words or phrases used in unusual or restricted ways, including idioms and jargon. This Success Criterion 3.1.3 (Unusual Words) is included in Guideline 3.1 (Readable), which recommends making text content readable and understandable. Likewise, this guideline belongs to Principle 3 (Understandable), which states that the user interface's information and operation must be understandable. As shown in Table 2, Success Criterion 3.1.3 requires that the definition of a word be provided when the word is used in an unusual or restricted way on a webpage. To follow the techniques and provide definitions for unusual words, it is necessary to differentiate between the following two situations: if a word has just one meaning within the webpage or if different meanings for the same word appear within the same webpage.

Furthermore, design pattern 4.4.1 of the COGA documentation indicates that common and clear words must be used in all content. Some techniques add a simple language term and provide a definition if complex words are used.

To apply these techniques, one must (1) detect which words are unusual or complex; (2) offer simpler synonyms; (3) offer

TABLE 1. Readability and Understandability Guidelines.

GUIDELINES	DESCRIPTION
WCAG 2.1 [5]	✓ ***Unusual Words*** (Success Criteria (SC) ***3.1.3)*** A mechanism is available for identifying specific definitions of words or phrases used in an unusual way. • Abbreviations (SC 3.1.4) A mechanism for identifying the expanded form or meaning of abbreviations is available. • Reading Level (SC 3.1.5) When the text requires advanced reading ability, a reading version is available that does not require advanced reading ability.
Easy-to-Read [7]	✓ ***Do not use difficult words*** • Speak to people directly. • Use positive sentences rather than negative ones where possible. • Use active language rather than passive language where possible. • Use personal pronouns.
Plain Language [9][10]	✓ ***Write in clear and easily understandable language.*** • Use short, concise sentences. • The content can be repeated. Explain difficult words. Give examples. • Do not write in metaphors. Do not use abbreviations or explain them if you do.
COGA documents [12]	✓ ***4.4.1 Pattern: Use Clear Words.*** • 4.4.2 Pattern: Use a Simple Tense and Voice. • 4.4.3 Pattern: Avoid Double Negatives or Nested Clauses. • 4.4.4 Pattern: Use Literal Language. • 4.4.5 Pattern: Keep Text Succinct. • 4.4.6 Pattern: Use Clear, Unambiguous Text Formatting and Punctuation. • 4.4.8 Pattern: Provide Summary of Long Documents and Media.

TABLE 2. Success criterion 3.1.3 techniques (WCAG 2.1) [5].

Situation A: If the word or phrase has a unique meaning within the web page:	
G101: Providing the definition of a word or phrase used in an unusual or restricted way	G55: Linking to definitions • H40: Using description lists • H60: Using the link element to link to a glossary G112: Using inline definitions • H54: Using the *dfn* element to identify the defining instance of a word G62: Providing a glossary G70: Providing a function to search an online dictionary
Situation B: If the word or phrase has different meanings within the same web page:	
G101: Providing the definition of a word or phrase used in an unusual or restricted way	G55: Linking to definitions • H40: Using description lists • H60: Using the link element to link to a glossary G112: Using inline definitions • H54: Using the *dfn* element to identify the defining instance of a word

definitions; and (4) contextualize the meaning of the unusual word in the text to offer the correct synonym or definition.

The design of our proposal is based on these requirements. To implement (1) and (2), a lexical simplification approach is proposed to identify complex words, and once they are identified, all synonyms are generated and the simplest synonym given the context is selected (thus implementing (3)). Moreover, this system not only obtains definitions but also includes a disambiguation system to take into account the context (thus implementing (4)). These steps are described in Section IV.

III. RELATED WORK

A. NLP AND ACCESSIBILITY

NLP is a discipline dedicated to developing technology capable of understanding natural language in a way similar to human beings. One area in which this could be applied is the development of technology that improves accessibility for individuals with disabilities. An example of this is the implementation of simplification processes that transform a text into an equivalent but simpler version for people with intellectual disabilities.

Regarding accessibility, works that focus on simplification geared towards groups of individuals with disabilities were found, such as [13], which provided text simplification for deaf users by providing syntactic and lexical paraphrasing of a text to assist in the comprehension of the text's meaning. Additionally, there are works focusing on lexical simplification for people with autism [14]–[16], aphasia [17], [18], low vision [19] or dyslexia [20], [21], who could have comprehension problems. Finally, Simplext [22] and the works introduced in [23], [24] offer automatic text simplification in Spanish for individuals with intellectual disabilities.

B. NLP APPROACHES TO LEXICAL SIMPLIFICATION

In 1996, the first automatic simplification approaches [25] provided a superficial analysis of texts to identify verbs and nouns in complex phrases. Among the many ways of approaching this task, syntactic simplification consists of identifying grammatical complexities and converting them into much simpler versions [26]. Lexical simplification involves substituting words in a given phrase to make it simpler, without modifying its syntactic structure in any way. Different methods are used to accomplish this task, from supervised machine learning (ML) algorithms to unsupervised ML algorithms or even the recently proposed hybrid approaches [27]. Supervised approaches require annotating datasets to achieve their objective [28], which leads to a significant disadvantage when dealing with languages with few annotated corpora for text simplification [22], [29], as is the case of Spanish. While unsupervised approaches outperform supervised approaches in terms of coverage, they have the disadvantage of only performing one-to-one substitutions and cannot deal with phrases. They also tend to change the meaning of the sentence and have problems dealing with ambiguous words [30]–[32]. However, recently, unsupervised approaches have been improved in this regard by allowing more detailed context information to be obtained [33]. On the other hand, hybrid strategies employ methods from both the previous two approaches, such as [34], which uses a corpus-based approach and a combination of a free lexicon, decision trees, and context-based rules.

Among the methodological approaches, the most recently researched ones are data-driven approaches. They can simultaneously perform multiple simplification transformations and learn very specific rewrite patterns and complex rewriting patterns using rule-based approaches, such as rules for splitting and reordering sentences from large datasets [35]. Specifically, lexical simplification [36], [37] proposes four steps to achieve this goal, as follows: complex word identification (CWI), generation of substitutes, selection of substitutes and substitute ranking. This approach is followed in this work to detect complex words and offer simpler synonyms, but with the additional contribution of using Easy Reading and some state-of-the-art resources for text classification and definition search tasks, as described in Section IV.

CWI aims to select the candidates to be simplified, that is, to detect which words are complex in a given text. ML approaches have been shown to surpass other strategies. Shardlow [38] compared binary support vector machine (SVM), threshold-based, and "Simplify Everything" approaches, where in the latter, it is assumed that all words in a sentence can be simplified. The results demonstrated that the SVM approach outperforms the others in terms of precision. Recently, ML approaches have been used quite frequently in this task. In the BEA workshop [39], on the CWI task for uni/multiword phrase classification, most participant teams preferred ML approaches for their systems. For example, [40] presented three approaches for CWI, one using the traditional classification algorithms of ML based on lexical features (word length, number of syllables, and others) and N-gram features (probabilities of n-gram). Other works outside workshops have also been carried out, such as [41], which used the task dataset to train a convolutional neural network (CNN) with word embeddings and engineered features. Achieving an F1-score of 0.79, this approach outperformed the workshop participants' results. In this research work, we focused on finding a new combination of features of various types to outperform other systems in representing a word in the dataset of the described task, as supported by a linear SVM (see the results in Section V).

In relation to substitute generation, this second step involves producing substitute candidates for the complex words detected. The previous works followed two strategies, i.e., linguistic database querying and automatic generation [36]. The former, relying on linguistic databases manually constructed by professionals in which a target word has a number of synonyms or related words attached to it, was the used most often. Some examples of this strategy are WordNet [42], [43] and the OpenThesaurus database, which helped the work of [23] by providing synonyms for 21,381 target words in Spanish. While this strategy has the advantage of presenting a very sensitive and precise approach, it also has the disadvantage of not having broad coverage, especially in Spanish.

Furthermore, constructing these databases is a time-consuming task. Automatic generation focuses on overcoming this disadvantage and seeks to gather extracted candidates from less expensive resources by using, for example, regular word definition dictionaries, as in the case of [44], which combines this with a part-of-speech (POS) tagger to search for candidates with a similar POS tag pattern to the target word. Other approaches rely on parallel corpora in which simplified sentences are created by evaluating original simplification pairs [45]. A more straightforward approach was presented by [46], where they initially annotated a paraphrase dataset to train a model to classify simplified paraphrases, resulting in the simple multilingual Paraphrase Database (PPDB). This database contains over a billion paraphrases. Despite the scarcity of resources available that provide synonyms for Spanish, we followed a strategy that combines both approaches, i.e., querying the linguistic database and automatic generation. Subsequently, we used text cleaning techniques to obtain a more efficient approach, as shown in Section VI.

In the third step, in which a substitute is selected from the set of synonyms extracted from the previous step, the most suitable synonym is chosen according to its simplicity and the context. In this stage, the selected synonym should preserve the original meaning of the sentence, as well as provide a correct syntactic structure. Several strategies have been proposed in recent years, starting with the explicit sense labelling strategy, where the selection of a substitute is posed as a word sense disambiguation (WSD) problem [47]. However, this strategy has the disadvantage of requiring manually created sense/synonym databases that are expensive to produce. In an attempt to overcome this issue, the implicit sense labelling strategy was created by automating the learning of complex word meaning classes instead of using databases [48].

Moreover, in languages where WSD resources are sparse or unavailable, POS strategies were proposed, as in [49], where the words are filtered adhering to a specific set of rules, including among others, the POS tag of the candidate. This is done to ensure that the meaning of the original word is maintained. Unfortunately, this approach showed poor results when dealing with highly ambiguous words. Therefore, to address these problems, recent works incorporate similarity metrics in the selectors. In [50], the authors selected the final synonym using the cosine distance in a word embedding model. Given a word to be simplified, the word with the closest vector based on cosine similarity was chosen. In this work, the latter approach has been selected and optimized to evaluate the similarity between the target word and the context in which it is found in the sentence with Word2vec and BERT models. Subsequently, to refine these results, combinations of different generators and the best selector are evaluated. These experiments are detailed in Section VII.

As observed in the substitute selection step, WSD is extremely useful when dealing with word ambiguity. As new words continue to be added to our language, this task becomes increasingly complex. Furthermore, as it is influenced by the domain in which the knowledge is created, producing resources to support knowledge-based WSD [51] is extremely costly. Considering this disadvantage, ML approaches have proven to be good solutions through the use of predictive strategies. Supervised [52], unsupervised [53], and semisupervised [54] approaches have been used in the research. However, others have approached this problem from another point of view, such is the case of Google, which uses its BERT language representation model (bidirectional encoder representations from transformers) [55] to solve different NLP tasks by fine-tuning their pretrained models. A research project based on this approach [56] consisted of fine-tuning a BERT model for WSD using WordPiece (multilingual text tokenizer) embeddings as part of the entries. Good results were obtained, outperforming the results of current approaches with regard to their F1-scores. In this work, to meet the accessibility guidelines, a module that can provide definitions for complex words was needed. However, the problem regarding the polysemy present in the Spanish language quickly arose. While BERT has been shown to disambiguate textual content by having more context information than other approaches, it alone cannot determine which dictionary definition pertains to a specific word. Thus, we take a multilingual BERT model as a starting point and then create a process for solving word disambiguation when searching for a definition (detailed in Section VIII).

IV. LEXICAL SIMPLIFICATION SYSTEM

This section presents the EASIER system, which provides systematic support for compliance with the accessibility guidelines described in Section II. This support is offered by implementing a lexical simplification system that identifies complex words and proposes synonyms and definitions that provide the best fit while taking into account the context using NLP approaches.

Figure 1 shows the modular system architecture, indicating the NLP resources used in each module. The system follows a process based on a Lexical Simplification Pipeline according to the approach [36] previously introduced. The process starts by identifying unusual words (complex word identification module). The next step is to offer simpler terms (generation/selection of substitutes). Finally, to make the content more understandable, as the accessibility guidelines suggest that the definition of an unusual word should be provided and since many words in Spanish are polysemic, a word sense disambiguation module has been created.

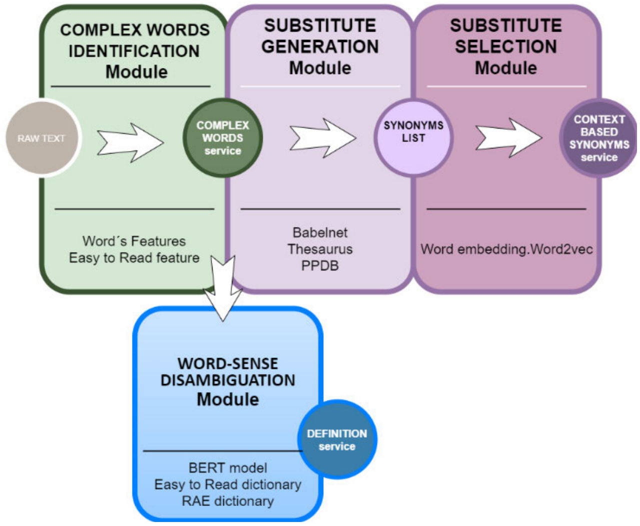

FIGURE 1. EASIER system description.

Each module, along with its evaluation, is described in the following sections.

V. COMPLEX WORD IDENTIFICATION MODULE

The CWI module follows an ML approach that requires datasets with words labelled as either complex or simple to train and validate our algorithm. These datasets are described as follows:

A. TRAINING DATASET

As in our previous work [57], to train/test our classifier, we use the datasets from the shared task of multilingual CWI from the BEA Workshop 2018, which can be found at sites.google.com/view/cwisharedtask2018. This dataset provides a list of words and their corresponding classification (1 for complex or 0 for simple) and additional useful information that can assist in the classification tasks.

The training dataset contains a total of 13,747 instances, of which 40% represent complex words, and the test dataset contains a total of 2,233 instances, of which 41% represent complex words. A more in-depth description of this task and dataset can be found in the workshop report [58].

Each instance contains a target uniword/multiword that is selected by annotators. Moreover, each instance is represented by 11 columns, which provide a range of information. The dataset contains binary information (labelling the target words in context as complex or simple) and probabilistic (assign the probability of target words in context being complex) subtasks. For the development of this system, we focus on the binary classification subtask and use the following information:

- The first column shows the ID of the sentence.
- The second column shows the actual sentence in which a complex phrase annotation exists.
- The third column shows the initial char offset of the target word in the sentence.

- The fourth column shows the end Char offset of the target word in the sentence.
- The fifth column shows the target word.
- The sixth and seventh columns show the number of native annotators and the number of nonnative annotators who examined the sentence.
- The eighth and ninth columns show the number of native annotators and the number of nonnative annotators who marked the target word as difficult.
- The Tenth Column shows the gold-standard label for the binary task (0: simple and 1: complex).
- The Eleventh Column shows the gold-standard label for the probabilistic task.

We do not consider the information in some columns (the sixth, seventh, eighth, ninth and eleventh columns) because the information in these columns is intended for use in the probabilistic task.

B. PROPOSED FEATURES

With this dataset and for the purposes of training the algorithm, each word (instance) must be represented as a set of features that help to distinguish between complex and simple words. As a first step in the evaluation and to find the best possible combination of features for this task, an analysis was performed with several features available for the Spanish language, which are described in the following section. Below, we describe the best proposed combination of features in this work:

- **Length feature**: word length.
- **Boolean feature**: if a word is composed of capital letters.
- **Word embedding (Word2vec) feature**: for each word, we extract vectors from a Word2vec model trained on The Spanish Billion Words Corpus [59].
- **Word embedding (BERT) feature - BERT model vectors**: to obtain these vectors, we use a multilingual, pretrained BERT model released by Google (www.github.com/shehzaadzd/pytorch-pretrained-BERT). To import this model, we use the PyTorch interface for BERT by Hugging Face (https://github.com/huggingface/transformers). The next step involves fine tuning the model specifically for our classification task, which results in a four-dimensional object for each sentence. This includes the layer number (12 layers), the batch number (1 sentence), the number of tokens and the feature number (768). To create our word vectors, we combine certain layer vectors by performing a concatenation of the model's last four layers. For this classification, we select a certain number of dimensions of the model (480). These dimensions are selected because they have shown better results in our tests, and it is worth mentioning that, depending on the task, this strategy could have different results.
- **E2R feature**: as an added value to other related work, we use resources from the domain of easy-to-read (E2R). We propose a new feature with the creation of an E2R dictionary. For each word, if a target word exists in the E2R dictionary, it is classified as 0; otherwise, it is designated as 1. The dictionary is fed from a range of sources that provide E2R literature developed by experts. Subsequently, this text is cleaned to preserve only the content words (noun, verbs, adjectives, adverbs). Presently, this dictionary contains 13,400 simple words.

C. EVALUATION

To validate our algorithm, we use the test dataset described in Section A. A different choice was made with regard to the SVM kernel in the CWI stage. In our previous work [57], we used the radial basis function (RBF) kernel. However, for this work, a linear kernel was used instead, which is much faster [60] and has the additional advantage that SVM has shown good performance in classifying sparse instances [61]. Concerning the typical metrics for this task, we use precision, recall, accuracy and the F1-score.

- **Precision:** the proportion of correct positive predictions.
- **Recall:** the proportion of actual positives correctly identified.
- **Accuracy**: the proportion of correct predictions to the total number of input samples.
- **F1-Score**: the harmonic average between precision and recall.

These metrics will help us determine if the classifier can avoid making unnecessary replacements of simple words and, therefore, make the sentence as simple as possible.

Table 3 shows the results obtained regarding the Train and Train/Developer datasets, which were validated with the test dataset. The results in both cases outperformed the results obtained by other systems from the abovementioned workshop [57].

TABLE 3. CWI results in test dataset.

	Accuracy	Precision	Recall	F1 Score
TRAIN	0.80	0.79	0.78	**0.792**
TRAIN+DEV	0.80	0.80	0.79	0.794

Additionally, to complement the previous information, Table 4 shows the scores of some combinations of features, determining which features are more discriminatory. These features can be grouped into length features (word length, sentence length and number of syllables), probability features (with window lengths of 1, 2 and 3), Boolean features (word morphology), embedding features (Sense2vec, Fasttext, Word2vec, BERT) and the E2R feature. One of the best scores is reached with the help of the embedding model vectors. Using Word2Vec and the BERT models, an F1-score of 0.752 is obtained. Furthermore, when evaluating the F1-scores independently for each feature, the Word2Vec feature yields a score of 0.70, which proves to be a valuable resource for this task. The BERT feature shows an F1-score of 0.727, which is the best score achieved among all independent features.

TABLE 4. CWI results of feature combinations, where WL: Word Length, SN: Syllable number SL: Sentence Length, P: Probability, E: E2R, F: Fasttext, W2V: Word2vec, S2V: Sense2Vec, and BT: BERT.

Feature	Accuracy	Precision	Recall	F1 Score
WL	0.73	0.74	0.70	0.702
SN	0.724	0.719	0.695	0.700
SL	0.593	0.296	0.5	0.372
P	0.716	0.709	0.689	0.693
E	0.593	0.296	0.5	0.372
F	0.699	0.659	0.589	0.569
W2V	0.72	0.71	0.70	0.700
S2V	0.797	0.797	0.777	0.783
BT (400 dimensions)	0.737	0.730	0.717	0.720
BT (450 dimensions)	0.742	0.735	0.720	0.725
BT (480 dimensions)	0.74	0.74	0.72	0.727
WL+SN	0.733	0.749	0.693	0.698
SN+SL	0.733	0.744	0.695	0.700
WL+SN+SL	0.735	0.748	0.697	0.702
WL+SN+SL+P	0.749	0.768	0.710	0.716
WL+SN+SL+B+E+W2V+F	0.790	0.789	0.771	0.776
W2V+BT	0.77	0.76	0.75	0.752
WL+BT	0.79	0.78	0.77	0.778
WL+B+BT	0.79	0.79	0.78	0.783
WL+B+E+BT	0.80	0.80	0.78	0.787
WL+B+E+W2V+BT*	**0.80***	**0.80***	**0.79***	**0.794***
WL+B+E+W2V+S2V+BT	0.802	0.797	0.788	0.791
WL+B+E+S2V+BT	0.797	0.794	0.779	0.784

The discarded features resulted in a lower score. Some of these were dropped due to several reasons, one of the most common being related to the available vocabulary of the feature, for example, in the probability feature, which independently scored 0.69. In many cases, the target word was not found in the resource vocabulary, which resulted in a vector full of null values. Another case was among the embedding features. First, it was believed that the combination of different embeddings would result in a higher score; however, this was not the case. We believe that these negative results occur because the models were created with different resources in the case of Sense2vec,[1] Fasttext[2] or BERT,[3] and consequently, each model presented different vocabularies and vectors, confusing the classifier. Further information regarding our evaluation can be found at github.com/ralarcong/EASIER_EVALUATIONS.

In addition, to evaluate our results compared to those obtained by other systems, Table 5 shows a comparison between our system and the seven best results from the BEA Workshop for the Spanish CWI task. With the training dataset, our system outperforms the other systems, obtaining a score of 0.792. Our system is ranked directly above TMU [62]. This system is based on the frequency of the target word in a Wikipedia Corpus and a learner corpus subsequently trained on a random forest classifier, as well as a deep learning architecture with word/char embeddings, word length and frequency counts named NLP-CIC [63].

[1] https://github.com/explosion/sense2vec
[2] https://github.com/facebookresearch/fastText
[3] https://github.com/shehzaadzd/pytorch-pretrained-BERT

TABLE 5. F-1 scores for the CWI task from the BEA Workshop 2018.

Systems for the Spanish Language	F-1
Our approach	**0.792**
TMU	0.769
NLP-CIC	0.767
ITEC	0.763
NLP-CIC	0.746
CoastalCPH	0.745
CoastalCPH	0.745
NLP-CIC	0.741

VI. SUBSTITUTE GENERATION MODULE

The substitute generation module generates substitution candidates for complex words, considering all the contexts in which they may appear.

A. PROCEDURE

We test the performances of different substitute generation strategies by using the resources mentioned above and applying rules to search for a better result. In this step, we extract substitutes for a target word from a variety of linguistic resources. Table 6 shows the resources used by the generator and the selector components.

The generators tested are as follows:

TABLE 6. Resources for substitute generation/substitute selection.

Resource	Description	Webpage
Thesaurus	Synonym database	thesaurus.altervista.org/
Babelnet	Synonym database	live.babelnet.org/
PPDB	Paraphrase database	paraphrase.org/
BERT	BERT pretrained model	github.com/shehzaadzd/pytorch-pretrained-BERT
Word2Vec	Word2Vec pretrained model	crscardellino.github.io/SBWCE/

- Thesaurus database (named 1): synonym search for the target word.
- Thesaurus database (2): search for synonyms for the target word and its lemma.
- Babelnet database (3): search for synonyms for the target word.
- Babelnet database (4): search for synonyms for target word and its lemma.
- PPDB (5): search for replacements for the target word.
- PPDB (6): search for replacements for target word and its lemma.
- Babelnet + Thesaurus (7): concatenate the extracted values from (2) and (4).
- Babelnet + Thesaurus + PPDB (8): concatenate the extracted values from (2), (4) and (6).

Additionally, we evaluate the performances of the highest-ranked combinations by performing the cleaning techniques described below:

- Babelnet + Thesaurus (9): in addition to the procedure described in (7), we extract the target word's lemma and stem. Subsequently, we delete the candidate words that contain the stem or match the extracted lemma.
- Babelnet + Thesaurus + PPDB (10): in addition to the procedure described in (8), we extract the target word's lemma and stem. Subsequently, we delete candidate words that contain the stem or match the extracted lemma.

B. EVALUATION

A portion of the EASIER dataset, created within this research framework, is used as the gold standard (further information can be found at github.com/LURMORENO/EASIER_CORPUS). This portion[4] comprises 500 instances, each containing a sentence, a target complex word and three context-aware substitutions suggested by an expert linguist. The evaluation metrics used are those found in the work of Paetzold [35], which are as follows:

- **Potential**: the proportion of instances for which at least one of the candidates generated is contained within the gold standard.
- **Precision**: the proportion of generated substitutions that are contained within the gold standard.
- **Recall**: the proportion of gold-standard substitutions that are among the generated substitutions.
- **F-Score**: the harmonic average between precision and recall.

Table 7 includes the results obtained (see github.com/ralarcong/EASIER_EVALUATIONS). This step aims to achieve the highest coverage possible in the gold standard. Combining Thesaurus, Babelnet, and the PPDB resources seems to have the best performance when attempting to achieve this goal by reaching the highest potential and recall, which are 0.898 and 0.597, respectively. However, this strategy results in a low precision rate because of the higher number of false positives.

To offset this, it is worth mentioning that the cleaning techniques used in (9) and (10) had minimal impacts on the potential and recall but increased the precision score. By analysing this increase, it seems that, in some cases, the resources are outputting the target word in a different grammatical form and, consequently, providing false positives when evaluated with the gold standard.

TABLE 7. Substitute generation results.

	Potential	Precision	Recall	F1
(1)	0.288	0.047	0.170	0.074
(2)	0.5	**0.070**	0.248	**0.109**
(3)	0.312	0.042	0.156	0.066
(4)	0.760	0.051	0.426	0.091
(5)	0.796	0.048	0.480	0.087
(6)	0.808	0.050	0.485	0.090
(7)	0.644	0.059	0.335	0.099
(8)	**0.898**	0.043	**0.597**	0.080
(9)	0.890	0.060	0.564	**0.109**
(10)	0.896	0.054	0.589	0.098

The module demonstrated acceptable results by obtaining a potentially high rate and satisfactory recall rate, thus achieving this stage's main objective, which was to obtain possible replacements for a target word in all possible contexts.

VII. SUBSTITUTE SELECTION MODULE

The substitute selection stage takes the list of synonyms extracted from the previous step and selects the most suitable synonym according to its simplicity and the context.

A. PROCEDURE

As the core resource in this step, we use a word embedding model, where words are represented as numerical vectors in low dimensional space, supported by a word2vec similarity function, which allows us to calculate the cosine distance between word vectors. To obtain these similarities, we employ a pretrained Spanish Billion Words Corpus Word2vec embedding model. The selectors are as follows:

- **No selections** (named 1): selects all candidates.
- **Any Window** (2): obtains three similarity values (candidate and target word, candidate and target word's context words in the sentence (previous and subsequent words)). Next, these values are added and stored. Finally, this

[4]http://dx.doi.org/10.17632/ywhmbnzvmx.2

process is repeated for every candidate, and the selector picks the three candidates with the highest values.

- **Lexical Window** (3): Similar to (2), but instead of selecting the first context word, we select the first word with lexical content (previous and subsequent words).
- **CWI Model** (4): Before performing the selection, we filter the candidate list, excluding the complex words predicted by the CWI model observed in Section V. Then, the same process described in (3) is performed.

B. EVALUATION

We use the same dataset and metrics as in Section VI (see github.com/ralarcong/EASIER_EVALUATIONS). Since the selector needs candidates, we use the generators with the best potential ranking from the previous step (8). Thus, we can easily determine the selector's effectiveness based on which selector results in the largest number of potential right answers.

Table 8 illustrates the results. As expected, not performing any selection (1) yields the highest potential and recall scores. On the other hand, by extracting the first words in the context of the candidate, (2) shows an even lower score in every metric. However, by extracting the first words in the context with lexical content (3), we obtained better precision and F1 scores.

TABLE 8. Substitute selection results – GENERATOR (8).

	Potential	Precision	Recall	F1
(1)	0.**896**	0.054	**0.589**	0.098
(2)	0.022	0.008	0.006	0.007
(3)	0.406	**0.172**	0.121	**0.142**
(4)	0.006	0.003	0.001	0.002

Additionally, to improve the selector, we evaluate a combination of the highest precision ranked selector (Lexical Window (3)) and the previous generators described in Table 7 (shown in Table 9). Whereas our last generator and selector combination presented a better result, we found that helping the selector by filtering certain words presented even greater results (9).

Furthermore, to test one of the functionalities that the BERT model has to offer (11), we evaluate the prediction function to retrieve possible substitutions for a target word. The results showed very low performance since the BERT model suggested generic words, following the main objective of substitute generation but working against the main objective of substitute selection.

Table 10 shows an example for the word "Prevenir" *("to prevent")*. These are the results when no filter is used; the candidate list suggests the target word in different grammatical forms as candidates. Since our word2vec model evaluates similarity, the model is inclined to choose words with similar semantics. Thus, we can assist the model in processing different candidates for the target word by filtering these words.

The module obtained acceptable results by performing a combination of strategies with the generator and the selector, achieving greater precision than the generator. This is an important metric when selecting a correct replacement for a target word. Additionally, when analysing one of the best studies that reviewed the related work in the English language [36] and despite the difference in the number of resources for the language, it can be observed that the results are comparable to those of this work. Table 11 shows the best results for English reported in [36]. For the generators, the best results were obtained with the strategy of combining the different resources available for the language, and potential and recall values of 0.996 and 0.358, respectively, were obtained. On the other hand, for the selectors, a replicated version of De Belder's approach [48] obtained the best accuracy, which was 0.257; however, Paetzold's approach, which used the semantic similarity of the word embeddings, provided an overall higher F-1 score of 0.245.

TABLE 9. Substitute selection results – different generators.

	Potential	Precision	Recall	F1
(1)	0.234	0.111	0.090	0.10
(2)	0.368	0.174	0.122	0.144
(3)	0.226	0.167	0.086	0.095
(4)	0.376	0.164	0.113	0.134
(5)	0.382	0.165	0.114	0.135
(6)	0.392	0.168	0.116	0.137
(7)	0.36	0.157	0.114	0.132
(8)	0.406	0.172	0.121	0.142
(9)	**0.504**	**0.226**	**0.154**	**0.183**
(10)	0.502	0.222	0.153	0.181
(11)	0.216	0.089	0.063	0.074

TABLE 10. Examples of substitute selection output.

	Target Word	Select Synonyms
Filter off	Prevenir *(to prevent)*	Prevenir *(to prevent)* Preveer *(Foresee)* Prevenido *(Prevented)*
Filter on	Prevenir *(to prevent)*	Preparar *(to prepare)* Precaver *(Precaution)* Advertir *(Warn)*

TABLE 11. English substitute generation/selection system results.

	Potential	Precision	Recall	F1
Paetzold-NE (SG)	0.876	**0.310**	0.142	**0.195**
All (SG)	**0.996**	0.071	0.358	0.119
Our Approach (SG)*	0.898	0.043	**0.597**	0.080
Belder (SS)	0.408	**0.257**	0.046	0.078
Paetzold-BR (SS)	0.972	0.231	0.261	**0.245**
Our Approach (SS)*	0.504	0.226	0.154	0.183

VIII. WORD-SENSE DESAMBIGUATION (WSD) MODULE

This section introduces the WSD module to select the correct definition for a specific word [64]. The core module uses a multilingual, pretrained BERT model released by Google, as described in the CWI module in Section V.

A. PROCEDURE

Figure 2 shows how the WSD procedure works where the target word considered is "exemplify". The following two dictionaries are used: the ''Real Academia de la Lengua Española'' Dictionary (RAE) (www.rae.es) and the ''Diccionario Facil'', with the latter being a dictionary of Easy Reading definitions created by the ''Plena Inclusión Madrid'' (www.diccionariofacil.org) association's experts and users with cognitive disabilities.

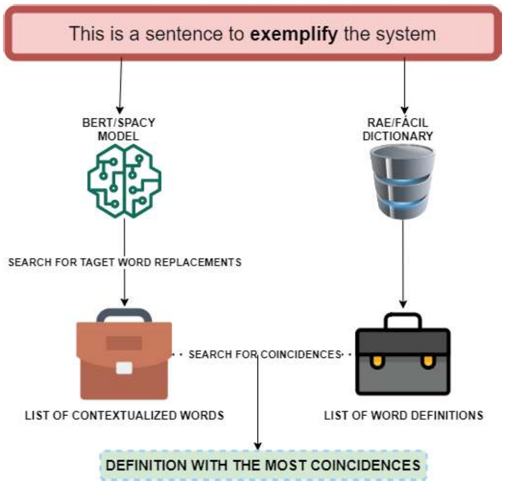

FIGURE 2. WSD procedure.

The system creates a list of definitions for the target word extracted from the RAE and Easy Dictionary. With the help of the model in the system, the word is masked in the sentence to which it belongs, and then the model predicts which words can be substituted for the masked word. This results in a list of words that share a common meaning, thus disambiguating the target word. With the help of Spacy (www.spacy.io/), these words are lemmatized to enrich the list. The words in the sentence with lexical content are then extracted and added to the list.

Since the first list created by our system contains words with similar semantics, these two lists are compared, and the coincidences are counted. The hypothesis followed is that the definition provided by the second list, which has more coincidences of words than the first list, is the correct definition associated with the target word and, consequently, is chosen by the WSD system (called the BERT approach). If no coincidences are found, the system selects the first definition on the list (called the first-in approach).

B. EVALUATION

The WSD module evaluation was performed by an expert linguist specializing in Easy Reading and Plain Language (see github.com/ralarcong/EASIER_EVALUATIONS). The expert received 525 sentences associated with the target word and the definition selected by the system. The expert verified whether the definition selected by the system was correct, taking the context of the word in the sentence into consideration.

As shown in Table 12, the BERT model approach was able to process 117 instances, with 64.95% rated as correct. By applying the ''First In'' strategy, 408 instances were processed, of which 72.06% were rated as correct. These results demonstrate that our approach performs well when dealing with polysemic words but faces some issues on recall.

TABLE 12. WSD module results.

Approaches	# Instances	% Correct
BERT Model	117	64.95
First In	408	72.06
Total	525	70.48

IX. DISCUSSION

We present and discuss the results obtained by each of the evaluation procedures separately in the following three subsections.

A. CWI

The CWI module results reveal that the use of embedding models benefits the classifier in its decisions, obtaining satisfactory F-1 scores when evaluated independently (e.g., Fasttext: 0.569, Word2vec: 0.70, Sense2vec: 0.783, BERT: 0.727). The latter embedding is useful for semantic searches and information retrieval. The main difference between this type of embedding and others, such as Word2Vec or FastText, is that BERT produces word representations that are dynamically informed by the words around them (contextualized embeddings). In contrast, with word2vec, the words are represented as unique indexed values. In the common word embedding models, each word is represented with one single vector, and polysemic words are ignored. In a sense, each word could have several vectors, one for each of its possible meanings. Therefore, these models allow us to deal with the task of word disambiguation when identifying complex words.

Regarding the other features, we found that determining the word length was sufficient to represent the word in terms of length by independently obtaining a score of 0.70. In features such as probability or E2R, the size of the dictionary was significant and represented its first disadvantage by giving null values when a word was not found. However, the E2R feature was beneficial because despite its limited size, when combined with features such as length and embedding, the results showed a significant improvement, with a final F1 score of 0.794. This result surpasses the other systems' scores by giving more detailed word information at the contextual, semantic and morphological levels.

B. SG/SS

By definition, in the generation of substitutes stage, one has to look for substitutes for a word in any context that may appear. However, we faced a disadvantage regarding the scarcity of resources in the Spanish language. Therefore, we opted for the fusion of different resources available for this language and obtained a recall of 0.597 and a high potential of 0.898. When evaluating the resources independently, Thesaurus obtained a recall of 0.248, Babelnet achieved a score of 0.426 and PPDB obtained a score of 0.485. These results make sense because Thesaurus is a smaller resource, and Babelnet, despite having more coverage, offers fewer substitutes per word than PPDB.

On the other hand, for the substitute selection stage, precision is more important than recall when offering the most suitable replacements for a given context. While making no selection preserved the recall of the generator, it showed a very low precision (0.054). Then, when using Word2vec to evaluate the word similarity, a better score of 0.172 was obtained. However, there were still points to improve because the generator in some cases offered replacements that were the same target word in a different grammatical form. Consequently, the Word2vec model chose these words because of their greater similarity to the original word. To mitigate this disadvantage, text cleaning techniques were used to remove these from the generator, resulting in a final precision of 0.226.

In addition, when we compare our generators and selectors to the previous work performed for the English language, we achieve comparable results, which in general, suggests that there is still much room for improvement for these tasks, as this work is an important contribution for the Spanish language.

C. WSD

Searching for a definition of a complex word is not an easy task due to the great ambiguity present in the language. BERT was used to deal with the disambiguation of a word based on its context by providing other words that fit the same context. The module uses these words to find matches in the definitions extracted from the RAE and ''Diccionario Facil''. The results showed good accuracy of 0.704 when evaluating 525 sentences associated with a target word and definition; however, the coverage was low. It was observed that matches were missing, as some word definitions had a different grammatical form than the words provided by the WSD system. Additionally, the module presents problems when generic sentences are evaluated. In this case, it outputs generic words, therefore selecting incorrect definitions. Another issue found was that when the system encounters the same number of coincidences among definitions, it assigns the last processed definition.

X. EASIER WEB SYSTEM

The Lexical Simplification System has been integrated into a web platform that shows the suitability of the proposal (github.com/LURMORENO/easier). A Spanish text is entered by a user (see Figure 3), and complex words are identified. Synonyms, a definition, and a pictogram are offered for each complex word detected. Moreover, language and accessibility resources are used, such as an easy-to-read dictionary (see Figure 4).

Furthermore, following the Easy-to-Read guideline ''To illustrate your text, you can use: photographs, drawings, or symbols'' or the Plain Language guideline

FIGURE 3. Screenshot of the EASIER system user interface.

FIGURE 4. Screenshot of the results for the EASIER system user interface.

''Use pictorial representation and other media: as an illustration, as support while reading'', the EASIER platform provides a pictogram of the complex word. This is obtained through the ARASAAC resource website API (www.arasaac.org/developers/api), which offers graphic elements for people with communication disabilities.

Additionally, the EASIER platform has been designed to comply with WCAG 2.1 (Level AA). COGA guidelines have also been followed, such as using clear and understandable content and making each step of the simplification process as clear as possible, including instructions. Moreover, a consistent visual design using symbols that assist the user has been used.

The webpage's user interface has been designed responsively, and a user interface for mobile devices is also provided. Moreover, browser extensions have been developed for both Chrome and Mozilla browsers that offer the function of identifying complex words and providing synonyms for text users to select on any webpage using the EASIER system.

XI. CONCLUSION

People with intellectual disabilities and other groups face cognitive accessibility barriers when they read texts that contain complex words that are not common or familiar to them. In this regard, there are accessibility directives and guidelines that aim to improve the readability and understanding of texts. This research work, which consists of the design and development of a Lexical Simplification System, aims to systematically comply with these guidelines. The system identifies complex words and replaces them with simpler synonyms. Additionally, a definition that is easy to understand is provided.

For this purpose, a pipeline with different steps has been followed, including a CWI module in which complex words are identified with the help of state-of-the-art and accessibility resources. The results show that our approach achieves better F1-scores than other systems in the same task, achieving a score of 0.794.

Regarding the substitute generation/selection steps, when combining linguistic resources and embedding were acceptable, higher potential indexes were obtained for the generator and are compared with the results presented in English. However, there are elements to be improved to obtain better precision results.

Finally, to provide a correct definition for unusual polysemic words, a WSD system has been created that uses a context-aware approach supported by a multilingual BERT model. The system was evaluated by an expert linguist, obtaining a satisfactory precision score.

Future research lines to improve the performance of our system are to explore other types of classification approaches, such as recent deep learning approaches (e.g., graph-based neural networks). In addition, for unusual word detection and substitute selection, the rule-based strategies can be evaluated by using the frequency resources available for the language. On the other hand, the different functionalities of BERT should be explored.

REFERENCES

[1] *The Survey of Adult Skills*, OECD, Paris, France, 2016.

[2] R. Zelezny-Green, S. Vosloo, and G. Conole, "Digital inclusion for low-skilled and low-literate people: A landscape review," United Nations Educ. Sci. Cultural Organ., Geneva, Switzerland, Tech. Rep., 2018.

[3] OCDE. (2013). *Resultados del Informe PIAAC de la OCDE*. [Online]. Available: http://www.educacionyfp.gob.es/prensa/actualidad/2013/10/20131008-piaac.html

[4] Eurostat. (2020). People in the UE—Population projection. Statistics Explained. Accessed: Sep. 2020. [Online]. Available: https://ec.europa.eu/eurostat/statisticsexplained/

[5] W3C. (2019). *Web Content Accesibility Guidelines (WCAG)*. [Online]. Available: https://www.w3.org/WAI/standards-guidelines/wcag/

[6] L. Moreno and P. Martinez, "The harmonization of accessibility standards for public policies," *Computer*, vol. 52, no. 7, pp. 57–66, Jul. 2019.

[7] K. Smith, G. Hallam, and S. B. Ghosh. (2012). *Guidelines for Professional Library/Information Educational Programs-2012*. IFLA Educ. Train. Sect. IFLA, Hague. Accessed: Aug. 25, 2014. [Online]. Available: https://www.ifla.org/publications/guidelinesfor-professionallibra ryinformationeducational-programs-2012

[8] G. Freyhoff, G. Hess, L. Kerr, E. Menzel, B. Tronbacke, and K. Van Der Veken, "Make it simple, European guidelines for the production of easy-to-read information for people with learning disability for authors, editors, information providers, translators and other interested persons," Int. League Societies Persons Mental Handicap Eur. Assoc., Brussels, Belgium, Tech. Rep., 1998.

[9] Plainlanguage.gov. *Plain Language Action and Information Network (PLAIN)*. Accessed: Nov. 2, 2020. [Online]. Available: https://www.plainlanguage.gov/about/

[10] Plainlanguage. (2017). *Plain English-Free Guides (co.uk)*. [Online]. Available: http://www.plainenglish.co.uk/free-guides.html

[11] *Guidelines and Recommendations for the Elaboration of Documents*, Standard UNE 153101:2018, Spanish Association for Standardization, 2018. [Online]. Available: https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0060036

[12] W3C Working Draft 11. (2020). *Making Content Usable for People With Cognitive and Learning Disabilities*. [Online]. Available: https://www.w3.org/TR/coga-usable/

[13] K. Inui, A. Fujita, T. Takahashi, R. Iida, and T. Iwakura, "Text simplification for reading assistance," in *Proc. 2nd Int. workshop Paraphrasing*, vol. 16, 2003, pp. 9–16.

[14] E. Barbu, M. T. Mart, and L. A. Ure, "Open book: A tool for helping asd users' semantic comprehension," in *Proc. 2th Worshop Natural Lang. Process. Improving Textual Accessibility*, 2013, pp. 11–19.

[15] R. Evans, C. Orasan, and I. Dornescu, "An evaluation of syntactic simplification rules for people with autism," in *Proc. 3rd Workshop Predicting Improving Text Readability Target Reader Populations (PITR)*, 2014, pp. 131–140.

[16] E. Barbu, M. T. Martín-Valdivia, E. Martínez-Cámara, and L. A. Ureña-López, "Language technologies applied to document simplification for helping autistic people," *Expert Syst. Appl.*, vol. 42, no. 12, pp. 5076–5086, Jul. 2015.

[17] J. Carroll, G. Minnen, Y. Canning, S. Devlin, and J. Tait, "Practical simplification of English newspaper text to assist aphasic readers," in *Proc. AAAI Workshop Integr. Artif. Intell. Assist. Technol.*, 1998, pp. 7–10.

[18] S. Devlin and G. Unthank, "Helping aphasic people process online information," in *Proc. 8th Int. ACM SIGACCESS Conf. Comput. Accessibility (Assets)*, 2006, pp. 225–226.

[19] L. Sauvan, N. Stolowy, C. Aguilar, T. François, N. Gala, F. Matonti, E. Castet, and A. Calabrese, "Text simplification to help individuals with low vision read more fluently," in *Proc. Lang. Resour. Eval. Conf. (LREC)*, 2020, pp. 11–16.

[20] R. Wilkens, B. Oberle, and A. Todirascu, "Coreference-based text simplification," in *Proc. 1st Workshop Tools Resour. Empower People Reading Difficulties*, 2020, pp. 93–100.

[21] L. Rello, R. Baeza-Yates, S. Bott, and H. Saggion, "Simplify or help? Text simplification strategies for people with dyslexia," in *Proc. 10th Int. Cross-Disciplinary Conf. Web Accessibility (W4A)*, 2013, p. 15.

[22] H. Saggion, E. Gómez-Martínez, E. Etayo, A. Anula, and L. Bourg, "Text simplification in simplext: Making texts more accessible," *Proces. Leng. Nat.*, vol. 47, pp. 341–342, Sep. 2011.

[23] S. Bott, L. Rello, B. Drndarevic, and H. Saggion, "Can spanish be simpler? Lexsis: Lexical simplification for Spanish," in *Proc. COLING*, 2012, pp. 357–374.

[24] H. Saggion, "Automatic text simplification," *Synth. Lectures Hum. Lang. Technol.*, vol. 10, no. 1, pp. 1–137, 2017.

[25] M. Shardlow, "A survey of automated text simplification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 48, no. 1, pp. 58–70, 2014.

[26] I. G. Dios, "Análisis de la complejidad y simplificación automática de textos. El análisis de las estructuras complejas en euskera," *Sociedad Española para el Procesamiento del Lenguaje Natural*, pp. 155–158, Mar. 2017.

[27] S. Štajner, H. Saggion, and S. P. Ponzetto, "Improving lexical coverage of text simplification systems for Spanish," *Expert Syst. Appl.*, vol. 118, pp. 80–91, Mar. 2019.

[28] S. Stajner, I. Calixto, and H. Saggion, "Automatic text simplification for Spanish: Comparative evaluation of various simplification strategies," in *Proc. Int. Conf. Recent Adv. Natural Lang. Process.*, 2015, pp. 618–626.

[29] R. Mitkov and S. Stajner, "The fewer, the better? A contrastive study about ways to simplify," in *Proc. Workshop Automatic Text Simplification-Methods Appl. Multilingual Soc. Assoc. Comput. Linguistics*. Dublin, Ireland: Dublin City Univ., 2014, pp. 30–40.

[30] G. Glavas and S. Štajner, "Simplifying lexical simplification: Do we need simplified corpora?" in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 63–68.

[31] G. Paetzold and L. Specia, "Unsupervised lexical simplification for non-native speakers," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 3761–3767.

[32] S. Surya, A. Mishra, A. Laha, P. Jain, and K. Sankaranarayanan, "Unsupervised neural text simplification," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2058–2068.

[33] J. Qiang, Y. Li, Y. Zhu, Y. Yuan, and X. Wu, "Lexical simplification with pretrained encoders," in *Proc. AAAI*, vol. 34, no. 5, Apr. 2020, pp. 8649–8656.

[34] D. Ferrés, H. Saggion, and X. Gómez Guinovart, "An adaptable lexical simplification architecture for major Ibero-Romance languages," in *Proc. 1st Workshop Building Linguistically Generalizable NLP Syst.*, 2017, pp. 40–47.
[35] S. S. Al-Thanyyan and A. M. Azmi, "Automated text simplification: A survey," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–36, 2021.
[36] G. H. Paetzold and L. Specia, "A survey on lexical simplification," *J. Artif. Intell. Res.*, vol. 60, pp. 549–593, Nov. 2017.
[37] J. Bingel, G. H. Paetzold, and A. Søgaard, "Lexi: A tool for adaptive, personalized text simplification," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 245–258.
[38] M. Shardlow, "A comparison of techniques to automatically identify complex words," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics Proc. Student Res. Workshop*, 2013, pp. 103–109.
[39] S. M. Yimam, S. Stajner, M. Riedl, and C. Biemann, "Multilingual and cross-lingual complex word identification," in *Proc. Recent Adv. Natural Lang. Process.*, 2017, pp. 813–822.
[40] N. Hartmann and L. B. Santos, "NILC at CWI 2018: Exploring feature engineering and feature learning," in *Proc. 13th Workshop Innov. NLP Building Educ. Appl.*, 2018, pp. 335–340.
[41] K. Cheng, "Multilingual complex word identification: Convolutional neural networks with morphological and linguistic features," in *Proc. 12th Int. Conf. Recent Adv. Natural Lang. Process.*, 2019, pp. 83–89.
[42] P. Lal and S. Ruger, "Extract-based summarization with simplification," in *Proc. ACL*, vol. 10, 2002, pp. 3–10.
[43] J. Burstein, J. Shore, J. Sabatini, Y. Lee, and M. Ventura, "The automated text adaptation tool," in *Proc. Hum. Lang. Technol., Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2007, pp. 3–4.
[44] T. Kajiwara, H. Matsumoto, and K. Yamamoto, "Selecting proper lexical paraphrase for children," in *Proc. 25th Conf. Comput. Linguistics Speech Process. (ROCLING)*, 2013, pp. 59–73.
[45] O. Biran, S. Brody, and N. Elhadad, "Putting it simply: A context-aware approach to lexical simplification," in *Proc. 49th Annu. Meet. Assoc. Comput. Linguistics*, 2011, pp. 496–501.
[46] E. Pavlick and C. Callison-Burch, "Simple PPDB: A paraphrase database for simplification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2016, pp. 143–148.
[47] S. Thomas and S. Anderson, "WordNet-based lexical simplification of a document," in *Proc. KONVENS*, 2012, pp. 80–88.
[48] J. De Belder and M.-F. Moens, "Text simplification for children," in *Proc. SIGIR Workshop Accessible Search Syst.*, 2010, pp. 19–26.
[49] G. H. Paetzold and L. Specia, "Text simplification as tree transduction," in *Proc. 9th Brazilian Symp. Inf. Hum. Lang. Technol.*, 2013, pp. 116–125.
[50] G. Paetzold and L. Specia, "LEXenstein: A framework for lexical simplification," in *Proc. ACL-IJCNLP Syst. Demonstrations*, 2015, pp. 85–90.
[51] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine code from an ice cream cone," in *Proc. 5th Annu. Int. Conf. Syst. Documentation (SIGDOC)*, 1986, pp. 24–26.
[52] Y. K. Lee, H. T. Ng, and T. K. Chia, "Supervised word sense disambiguation with support vector machines and multiple knowledge sources," in *Proc. 3rd Int. Workshop Eval. Syst. Semantic Anal. Text (SENSEVAL)*, 2004, pp. 137–140. [Online]. Available: https://www.aclweb.org/anthology/W04-0834
[53] H. Kaji and Y. Morimoto, "Unsupervised word sense disambiguation using bilingual comparable corpora," in *Proc. 19th COLIN*, 2002, pp. 1–7. [Online]. Available: https://www.aclweb.org/anthology/C02-1058
[54] R. Cao, J. Bai, and H. Shinnou, "Semi-supervised learning for all-words WSD using self-learning and fine-tuning," in *Proc. 33rd Pacific Asia Conf. Lang., Inf. Comput. (PACLIC)*, vol. 33, 2019, pp. 356–361.
[55] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805
[56] J. Du, F. Qi, and M. Sun, "Using BERT for word sense disambiguation," 2019, *arXiv:1909.08358*. [Online]. Available: https://arxiv.org/abs/1909.08358
[57] R. Alarcon, L. Moreno, I. Segura-bedmar, and P. Martínez, "Lexical simplification approach using easy-to-read resources," *Procesamiento Del Lenguaje Natural*, vol. 63, pp. 95–102, Sep. 2019.
[58] M. Yimam, C. Biemann, S. Malmasi, G. Paetzold, L. Specia, S. Štajner, A. Tack, and M. Zampieri, "A report on the complex word identification sharedtask 2018," in *Proc. 13th Workshop Innov. NLP Building Educ. Appl.* New Orleans, LA, USA: Association for Computational Linguistics, 2018, pp. 66–78, doi: 10.18653/v1/W18-0507.
[59] C. Cardellino, "Spanish billion words corpus and embeddings," Tech. Rep., Mar. 2016. [Online]. Available: https://crscardellino.github.io/SBWCE/
[60] I. Segura-Bedmar, C. Colón-Ruíz, M. Á. Tejedor-Alonso, and M. Moro-Moro, "Predicting of anaphylaxis in big data EMR by exploring machine learning approaches," *J. Biomed. Informat.*, vol. 87, pp. 50–59, Nov. 2018.
[61] J. Kivinen, M. K. Warmuth, and P. Auerc, "The Perceptron algorithm versus Winnow: Linear versus logarithmic mistake bounds when few input variables are relevant," *Artif. Intell.*, vol. 97, no. 97, pp. 325–343, 1997.
[62] T. Kajiwara and M. Komachi, "Complex word identification based on frequency in a learner corpus," in *Proc. 13th Workshop Innov. NLP Building Educ. Appl.* 2018, pp. 195–199.
[63] D. De Hertog and K. U. Leuven, "Deep learning architecture for complex word identification," in *Proc. 13th Workshop Innov. NLP Building Educ. Appl.* 2018, pp. 328–334.
[64] R. Alarcon, L. Moreno, and P. Martínez, "Word-Sense disambiguation system for text readability," in *Proc. DSAI 9th Int. Conf. Softw. Develop. Technol. Enhancing Accessibility Fighting Info-Exclusion)*, 2020, pp. 1–6.

RODRIGO ALARCON received the master's degree in computer science and technology from the Universidad Carlos III of Madrid (UC3M). He is currently pursuing the Ph.D. degree with the HULAT Research Group, Computer Science Department, Universidad Carlos III de Madrid.

He is currently a Computer Engineer with the Universidad Católica de Santa María (Arequipa-Perú). He investigates how to combine accessibility techniques with machine learning techniques to perform text simplification tasks using plain language and easy reading resources. His research works are aimed to improving the understanding of texts for people with intellectual disabilities by following standards of cognitive accessibility. His research interests include natural language processing and accessibility.

LOURDES MORENO received the B.Sc. degree in mathematics from the Complutense University of Madrid, Spain, and the Ph.D. degree in computer science from the Universidad Carlos III de Madrid (UC3M), Spain.

She is currently an Associate Professor and a Researcher with the Human Language and Accessibility Technologies Group (HULAT), Computer Science Department, UC3M. Her research combines human–computer interaction with natural language processing using interdisciplinary approaches. Her research interests include ICT accessibility aimed at users with disabilities across a range of fields (education, public administrations, and health). She is the Chair of the Human–Computer Interaction Association (AIPO), Spain, a member of the General Council of the Spanish Computer Sciences Society (SCIE), and the Vice-Chair of CHISPA, the Spanish Chapter of the ACM-SIGCHI.

PALOMA MARTÍNEZ received the degree and Ph.D. degree in computer science from the Universidad Politécnica de Madrid, Spain, in 1992 and 1998, respectively. She is currently the Head of the Human Language and Accessibility Technologies Group (HULAT), Computer Science and Engineering Department, Universidad Carlos III de Madrid. She has been a principal investigator and participated more than 40 national and international research projects. She has coauthored more than 40 articles in indexed journals and more than a 100 international conference contributions. Her research interests include human language technologies, with a focus on information extraction in the biomedical domain and web accessibility. She is a currently a member of the Spanish Society for Natural Language Processing (SEPLN) and the Dynamization Network for Activities on Natural Language Processing Technologies. She is also a Collaborator of the Spanish Centre of Captioning and Audiodescription (CESyA).