Lexical Complexity Prediction: An Overview

KAI NORTH and MARCOS ZAMPIERI, George Mason University, USA
MATTHEW SHARDLOW, Manchester Metropolitan University, UK

The occurrence of unknown words in texts significantly hinders reading comprehension. To improve accessibility for specific target populations, computational modeling has been applied to identify complex words in texts and substitute them for simpler alternatives. In this article, we present an overview of computational approaches to lexical complexity prediction focusing on the work carried out on English data. We survey relevant approaches to this problem which include traditional machine learning classifiers (e.g., SVMs, logistic regression) and deep neural networks as well as a variety of features, such as those inspired by literature in psycholinguistics as well as word frequency, word length, and many others. Furthermore, we introduce readers to past competitions and available datasets created on this topic. Finally, we include brief sections on applications of lexical complexity prediction, such as readability and text simplification, together with related studies on languages other than English.

CCS Concepts: • **Reference works → Surveys and overviews;**

Additional Key Words and Phrases: Complex word identification, lexical complexity prediction, NLP, lexical simplification, text simplification, assistive technologies

ACM Reference format:
Kai North, Marcos Zampieri, and Matthew Shardlow. 2023. Lexical Complexity Prediction: An Overview. *ACM Comput. Surv.* 55, 9, Article 179 (January 2023), 42 pages.
https://doi.org/10.1145/3557885

1 INTRODUCTION

Understanding the meaning of words in context is fundamental for reading comprehension. The perceived difficulty, hereafter referred to as *complexity*, of a target word within a given text varies widely among readers. With an increased demand for distance learning and educational technologies [107], research into automatically predicting which words are likely to cause comprehension problems is becoming a popular area of research [115, 147, 185]. Systems have been created to identify complex words that are difficult to acquire, reproduce, or understand for children [79], second-language learners [89], people suffering from a reading disability, such as dyslexia [131] or aphasia [35, 53], or more generally, individuals with low literacy [59, 175].

In Computational Linguistics and **Natural Language Processing** (**NLP**), the task of automatically recognizing complex words is most often achieved by training **machine learning** (**ML**) models. These ML models assign a complexity value to each target word within an inputted extract, sentence, or text that allows for the identification of complex words. This information can

Authors' addresses: K. North (corresponding author) and M. Zampieri, George Mason University; email: kn1473@rit.edu; M. Shardlow, Manchester Metropolitan University, UK.

0360-0300/2023/01-ART179 $15.00
https://doi.org/10.1145/3557885

Table 1. Example Extract with an Identified Complex Word from the CompLex Dataset [146]

Original:	**Folly**	is	set	in	great	dignity
Simplified:	**Foolishness**	is	set	in	great	dignity

The complex word and its simplified version are in bold.

then be used to improve downstream lexical and text simplification systems that provide simpler alternatives to aid reading comprehension. Take the extract shown in Table 1, for example.

An ML model trained to identify complex words would recognize the word "*folly*" within the original extract as being complex. Such models would come to this decision based on a number of engineered or inferred features. For instance, these models would likely consider the word "*folly*" as being archaic, as having a low frequency within everyday speech, as being unfamiliar to its target demographic or to a general populace, as being acquired later during adolescence, and so forth. Having identified "*folly*" as being complex, these models may then pass this information downstream so that it can be simplified to "*foolishness*" resulting in the simplification shown in Table 1. Readers may then use this simplification to better understand the meaning of the original sentence or the target word. Alternatively, the simplification of "*folly*" to "*foolishness*" may serve to improve machine translation, since *foolishness* is likely to have a more synonymous equivalent in a target language than "*folly*" [170]. Another use case of identifying complex words is for authorship identification, whereby identifying the number of complex words within a text can serve as a means of measuring vocabulary richness which has traditionally been used as a linguistic fingerprint, hence authorship marker [1].

The task of identifying complex words is commonly referred to as **Complex Word Identification** (**CWI**) [115]. In recent years, CWI has been extended to **Lexical Complexity Prediction** (**LCP**) [146, 148].[1] This survey introduces the reader to LCP by providing a comprehensive overview of LCP literature, with a particular focus on the work carried out in the last 10 years that has primarily dealt with English; however, research investigating other languages has also been included and their contributions acknowledged.[2]

This survey comes at a time of unprecedented demand for LCP research motivated by recent developments in education technology and accessibility, such as the widespread use of virtual learning platforms in distance learning [107]. It also comes at a time of diversification, with LCP interacting with other topics in NLP, such as machine translation [170] and authorship identification [1, 156]. To the authors' knowledge, this survey fills a gap in the current LCP literature. It provides new researchers, as well as those who are already familiar with the field, with the most up-to-date key references, main research questions, advancements, and baselines needed to develop LCP further.

This survey has the following structure. Section 2 gives prior definitions of complexity and explains what complexity is, what difficulty is in relation to complexity, and what is meant by the term *complex* in LCP literature. Section 3 briefly describes the origin of complexity prediction research within lexical simplification. Section 4 outlines the different types of lexical complexity prediction, ranging from comparative, binary, continuous, and personalized to predicting the complexity of multi-word and numerical expressions. It also discusses whether systems designed for a target demographic outperform those for a generic population as well as whether predicting the lexical complexity of multi-word expressions is advantageous for LCP (Sections 4.4.1 and 4.5.1).

[1] In this article, we will be using LCP as the overarching term and CWI specifically when we refer to the binary task of complexity prediction (Section 4) [146, 148].
[2] We hope that this article helps to encourage the ongoing development of LCP systems for other languages (Section 9.2).

Section 5 presents the evaluation metrics used to measure the performance of LCP systems, such as accuracy, precision, recall, F1-score, G-score, mean absolute error, mean squared error, Pearson's correlation, and Spearman's rank. Section 6 details the international competitions that challenged participating teams with the development of LCP systems: CWI–2016 [115], CWI–2018 [185], ALexS–2020, [192], and LCP–2021 [147]. Section 7 provides a historical overview of the models used for LCP, including feature engineering approaches, neural networks to state-of-the-art transformer-based models. It also describes the best linguistic features for predicting lexical complexity along with the effect including context has on LCP systems' performance (Sections 7.1.4 and 7.4.1). Section 8 demonstrates LCP's place within the text simplification pipeline and several of its use cases and applications. Section 9 gives an overview of the English datasets and resources used for LCP together with several studies that have investigated languages other than English. It also discusses whether transfer learning is possible for predicting lexical complexity across multiple languages (Section 9.2.7). Section 10 ends by briefly outlining the future of LCP research, including its future opportunities and challenges.

2 DEFINING COMPLEXITY

Within linguistics, there exist two approaches to defining complexity, when being used to describe the "*complexity*" of a target word: (1) absolute and (2) relative.

2.1 Absolute Complexity

Absolute complexity, otherwise known as objective complexity [47, 122], refers to a form of complexity that is established by the objective linguistic properties of a word [33, 122]. These linguistic properties include morpho-syntactic, semantic, as well as phonological factors that make a word appear to be complicated, advanced, or convoluted in comparison to a simpler alternative. For instance, having a high number of morphemes, the presence of derivational or inflectional affixes, having multiple meanings, or having multiple vowels or diphthongs, are all characteristic of absolute complexity [33, 114, 122].

un–believ–able *or* **engag–ed.** (1)

The words, "*unbelievable*" and "*engaged*" both contain two or more morphemes. The word "*engaged*" also has the diphthong eɪ within its first morpheme, /ɛnˈgeɪʤ/, which is known to cause production errors for language learners [109, 141]. "*Engaged*" also has multiple meanings, with one referring to the act of being involved in an activity, and another being pledged to be married [105]. When used in ambiguous contexts, polysemous words can be troublesome as they hinder a sentence's readability with an example being, "*Do you know if he is available as I think he is engaged?*" [95]. Words, such as "*unbelievable*" and "*engaged*," are therefore words with a high degree of absolute complexity since their linguistic properties make them hard to reproduce or understand.

2.2 Relative Complexity

Relative complexity, also know as agent-related complexity [47, 122] or simply referred to as "*difficulty*" [33], refers to a type of complexity that is informed by the individual experience or psycholinguistic factors of the individual. For instance, experiences such as the cognitive load, or demand, acquisition difficulty, along with an individual's level of familiarity associated with a particular word or typography, may determine a word's level of relative complexity [33, 114, 122].

LIKEABLE *or* **gothic** (2)

Table 2. Examples of Candidate Replacements Generated by an LS System

Target Word	Rank	Candidate Replacement
Folly	#1	Foolishness
	#2	Recklessness
	#3	Silliness
	#4	Craziness
	#5	Stupidity

Chen and Xiao [37] and Liu et al. [93] show that capitalized words are hard for Chinese learners of English to decipher and therefore are cognitively demanding. This is because they have less variance in their overall shape as well as less variance between the shape and size of their individual letters in comparison to words presented entirely in lowercase or Chinese characters which differ greatly in their form: LIKABLE versus likable or 讨人喜欢 (Mandarin for likable or popular). Words in reference to a particular art, culture, pop-culture, or historical group are also hard for second-language learners to acquire, especially if no cognate or similar cultural knowledge is available in their native language [157, 187]. Words such as "*LIKABLE*" and "*gothic*" are subsequently words with a high degree of relative complexity as factors more associated with the individual, such as typographical unfamiliarity or lack of cultural knowledge, make these words hard to decipher.

2.3 Complexity in LCP

Within LCP research, a more generalized notion of complexity is used. In most cases, the term "*complex*" is simply used as a "synonym for difficulty" [100] and is specifically applied to the word level, hereby referred to as lexical complexity or complexity. In this field of research, complexity therefore refers to the difficulty an individual may have in acquiring, understanding, or reproducing a particular target word, which is often a result of a target word's linguistic properties as well as factors belonging to the individual. Take the following words, for example:

unbelievable *or* **gothic**. (3)

Both "*unbelievable*" and "*gothic*" have been rated as having a neutral to high degree of lexical complexity within LCP research, regardless of the type of complexity they exhibit, be it either relative, absolute, or both [97]. As such, LCP adopts defining characteristics from absolute and relative complexity as determining a word's generalized level of complexity. This generalized notion of complexity is used throughout this article when referring to the prediction of lexical complexity.

3 ORIGIN OF COMPLEXITY PREDICTION

Predicting the lexical complexity of a target word originated as a sub-task of **lexical simplification** (**LS**) [155]. LS aims to replace complex words and expressions with simpler alternatives while maintaining the meaning of the original text as exemplified within Table 1 [115]. To achieve this, LCP is used by a LS system for two purposes: (1) to identify complex words that are in need of simplification, and (2) to rank the suitability of simpler alternatives.

Devlin and Tait [53] and Carroll et al. [35] were the first to adopt an LCP precursor within their LS systems' pipelines (see Table 2). They used WordNet [106] as well as Kučera-Francis's frequency norms, calculated using the Oxford Psycholinguistic Database [177], to rank their synonymous and simplified word candidates on what they believed to be their level of complexity. By doing so, their systems provided the most appropriate simplifications for their target complex words allowing for

Table 3. Example of a Sentence Annotated with Binary Complexity Values by a CWI Classifier Taken from the CompLex Dataset [146]

Extract:	**Folly**	is	set	in	great	**dignity**
Complexity Value:	1	0	0	0	0	0

Target words of interest are in bold.

the creation of easier to read texts for aphasic readers; LCP's place within the **text simplification** (**TS**) pipeline is described in greater detail within Section 8.2.

LS–2012 [155] is arguably the first shared task that contained an LCP element. It tasked five participating teams to design systems to "rank a set of [candidate] words, from the simplest to the most difficult" [152]. Participating teams took into consideration a variety of features to conduct complexity prediction. The most common of these features being simplified word candidates' frequency [11, 92, 152], n-grams [92, 152], morpho-syntactic characteristics including context [11, 77], and psycholinguistic properties [77].

4 TYPES OF COMPLEXITY PREDICTION

4.1 Comparative Complexity

Using LCP to rank words in terms of their complexity gives rise to a unique type of complexity prediction: comparative complexity. This type of complexity prediction provides a value that is used to distinguish whether a target word is more or less complex than another target word. As a result, comparative complexity prediction is most often found as a sub-task of LS, rather than its own stand-alone task [152, 155]. For instance, several studies [23, 77, 112, 117] have trained various models at comparative complexity prediction with the aim of improving LS (see Table 3).

Gooding et al. [66] investigated the effect that comparative judgment labeling had on inter-annotator agreement. They discovered that annotators tasked with ranking the complexity of several target words presented in context agreed more consistently on their chosen labels than compared with annotators tasked with purely identifying complex words without ranking. With a higher rate of inter-annotator agreement comes a higher quality of complexity label, since the true complexity of a target word is more likely to be captured. As a result, systems trained on such data, or that likewise make comparative judgments, can be highly effective at distinguishing between complex and non-complex words.

4.2 Binary Complexity

From 2012 to 2018, complexity prediction research primarily focused on binary complexity prediction. Binary complexity prediction is what is referred to as CWI. CWI is the task of assigning a target word with a binary complexity value of either 1, marking that word as complex, or 0, denoting that word as non-complex.

CWI is therefore unlike comparative complexity prediction as it purely identifies complex words rather than making comparative judgments or ranking the complexity of simplified word candidates.

Shardlow [142] was the first to treat CWI as a stand-alone task separate from LS. He experimented with a **support vector machine** (**SVM**) for CWI and detailed the construction of a binary CWI dataset (Section 9.1.1) together with the impact several features had on his CWI system's performance (Section 7).

CWI–2016 [115] was the first shared task that challenged teams directly with binary CWI. This shared task increased the popularity of complexity prediction research (Section 6.1). However, CWI's modeling as a binary classification task presented a few shortcomings during

Table 4. Example of Annotator Disagreement of the Complexity of Two Target Words Annotated Using a 6-Point Likert Scale

Annotations										
Target Word	**A**	**B**	**C**	**D**	**E**	**F**	**G**	Avg.	BC	BC Label
frontier	**3**	4	5	**3**	4	**3**	**3**	3.57	1	Complex
Milwaukee	3	4	4	5	3	2	N/A	3.5	0 or 1	Unknown

Annotations (or labels) ranged from very simple (1), moderately simple (2), simple (3), to complex (4), moderately complex (5), or very complex (6) with the issue being the even distribution between simple and complex labels. Target words, annotations, and complexity values were taken from the Word Complexity Lexicon [97]. **BC** refers to binary complexity value. Annotations of interest are in bold.

CWI–2016 [115]. The most notable is that CWI systems were unable to accurately and consistently classify target words on the decision boundary, being those words had an uncertain and often debated level of complexity [193].

4.2.1 Issue with Binary Complexity. Studies have demonstrated that since lexical complexity is subjective and dependent on an individual's experience and *a-priori* knowledge, binary CWI is prone to low inter-annotator agreement [97, 193]. Annotators from different demographics, such as first language or region, have different opinions of what classifies as a complex word, with perceived complexity often changing on an individual-to-individual basis [97, 193]. It is this disagreement on whether a word is either a complex or a non-complex word during the annotation process, that creates target words with an uncertain level of complexity that degrades CWI performance.

Take the words "*frontier*" and "*Milwaukee*" displayed in Table 4 as an example. Within the Word Complexity Lexicon [97] that used a 6-point likert scale ranging from very simple (1) to very complex (6), "*frontier*" was given non-complex labels by four annotators and complex labels by three annotators, whereas "*Milwaukee*" was labeled with non-complex and complex labels three times each, respectively. Averaging these words' labels, we are left with average complexity values that depict an uncertain, or neutral level of complexity given their proximity to the median threshold of 3.5. Converting these annotations to binary complexity values is therefore problematic. The target word "*frontier*" would no longer be considered as being neutral, but rather as being complex as its average complexity value is now over that of the median threshold. This is regardless of the fact that the majority of the labels assigned to "*frontier*" are non-complex. The target word "*Milwaukee*," on the other hand, is a word on the decision boundary, meaning that it can be labeled as either non-complex or complex by a CWI classifier even though its typography may be evidently complex to those whom are unfamiliar with North American loanwords or proper nouns.[3] Being trained on such examples that have been potentially mislabeled results in CWI systems misclassifying unseen target words. For instance, features used to distinguish non-complex words may be inevitably associated with complex words or vice-versa. This, in turn, hinders overall CWI performance [146, 193].

4.3 Continuous Complexity

LCP was introduced to deal with target words with an uncertain level of complexity along with target words on the decision boundary [97, 146]. Unlike CWI, LCP alternatively provides a continuous complexity value that is not used to assign a binary complex or non-complex label. Instead,

[3] It is important to mention that Maddela and Xu [97] may have attempted to avoid such neutral labeling by recruiting an uneven number of annotators, being 11 in total. However, not all annotators labeled each instance.

Table 5. Example of a Sentence Annotated with Binary Complexity Values (**BC**), and Continuous Complexity Values (**CC**) by a LCP Regressor Taken from the CompLex Dataset [146]

Extract:	**Folly**	is	set	in	great	**dignity**
BC	1	0	0	0	0	0
CC	0.57	0	0.18	0	0.15	0.42

Target words of interest are in bold.

LCP models complexity on a continuum with varying degrees of difficulty with which it then attempts to predict. For instance, it assigns target words with a complexity label ranging from very easy to very hard that are linked directly to certain thresholds: very easy (0), easy (0.25), neutral (0.5), difficult (0.75), or very difficult (1).

By modeling complexity on a continuum, LCP provides a more fine-grained representation of the complexity of a target word as it allows for the prediction of more than two levels of difficulty [146]. For example, the target word "*folly*" can be more accurately predicted as a neutral to difficult complex word, whereas the target word "*dignity*" is no longer incorrectly classified as being entirely non-complex (Table 5). An LCP system is thus a linear regressor rather than a binary classifier and for this reason can classify target words that were problematic for prior CWI systems, including such words as "*frontier*" and "*Milwaukee*."

It is worth mentioning that LCP was not the first to predict lexical complexity as a continuous value. Probabilistic complexity prediction was also a regression-based task. However, different from LCP, probabilistic complexity prediction used continuous complexity values to make binary predictions [185]. This meant that its continuous complexity values were not used to predict varying degrees of complexity like LCP, but rather to indicate the probability of that target word being either complex or non-complex. CWI–2018 [185] developed systems for binary as well as probabilistic complexity prediction. It was the first shared task that moved away from binary CWI and subsequently laid the foundations for what is known as LCP. CWI–2018 [185] is described in more detail within Section 6.2.

4.4 Personalized Complexity

Complexity prediction researchers have also been interested in personalizing lexical simplification [89, 163]. Lee and Yeung [89] argued that prior LCP systems are unable to account for "variations in vocabulary knowledge among their users" [89], including other forms of idiosyncrasies, such as cross-linguistic influence.[4] Several other researchers [89, 163, 195] have also suggested that the previous "one size-fits-all" approach to LCP fails to accurately model varying perceptions of lexical complexity and as a result, personalized CWI was introduced [89, 163, 195]. This approach creates personalized CWI systems that cater to the individual user or a specific target demographic. These systems are engineered with, or are built to learn, user demographic features that they use to make predictions on an individual basis. These demographic features may include language proficiency, "native language, race, job, age, ethnicity, or education" [89, 163, 195].

4.4.1 Is Personalized Complexity Prediction Worthwhile?

Personalized complexity prediction systems have been found to outperform LCP systems designed for a generic population when tasked with predicting the lexical complexities of a target demographic. Zeng et al. [195] discover

[4] Cross-linguistic influence being defined as the effects a bilingual speaker's first language (L1) has on their second language (L2) production, hence complexity assignment [182].

Table 6. Example Extract with an Identified and Simplified MWE Taken from the Complex Dataset [146]

Original:	**Folly**	is	set	in	great dignity
Simplified:	**Foolishness**	is	set	in	great dignity
MWE Simplified:	**Foolishness**	is	set	in	**pride**

Complex words are in bold.

that demographic features, such as native language, race, job, and so on, can improve CWI performance when predicting the complexity of medical terminology. Tack et al. [163] created a system designed to predict how well learners of French understood the meaning of a French word by incrementally training their system on features representative of their user's lexical competency. Lee and Yeung [89] and Tack [161] have since implemented personalized CWI models trained on language proficiency and native language [89, 161]. Both studies demonstrated their personalized CWI systems as outperforming their non-personalized baseline models. Tack [161] also included contextual features and found that her combined personalized and contextual model outperformed other models that did not take demographic or contextual features into consideration. Personalized complexity is therefore a promising area of complexity prediction research as it is seen to outperform more generalized approaches. Further details regarding a personalized LS dataset are presented in Section 9.1.4.

4.5 Multi-Word Expressions

LCP as well as other types of complexity prediction are not restricted to predicting the complexity values of single words. **Multi-word expressions (MWEs)** have also been studied and their complexity values predicted [147, 185]. However, there exists little research into the complexity prediction of MWEs.

4.5.1 Is Predicting the Lexical Complexity of Multi-Word Expressions Advantageous? According to Gooding et al. [67], assigning complexity values to both single words and MWEs would undeniably improve the performance of LCP systems and, as a consequence, the performance of other downstream NLP-related tasks, such as LS. Gooding et al. [67] provide "*ballot stuffing*" as an example. For instance, if complexity values were assigned individually to "*ballot*" and then to "*stuffing*," this MWE would either not be simplified, as individually "*ballot*" and "*stuffing*" may not be considered to be complex words, or simplified into an expression that would be "nonsensical or semantically different" [67], such as "*ballot filling*" or "*vote stuffing*." Another example can be seen in Table 6.

As shown in Table 6, "*great*" and "*dignity*," when taken into consideration separately, are not considered to be complex words. However, if these two words were presented to an annotator as one MWE, they may subsequently have been assigned a higher combined complexity value resulting in them as being identified as complex. In turn, a LS system may then provide a more appropriate simplification, such as "*pride*," that would further improve the readability of this extract. For this reason, the CompLex dataset [146] provides 1,800 MWEs with preassigned complexity values. LCP–2021 [147] was the first shared task that challenged teams to develop LCP systems to predict the complexity values of single words and MWEs as two separate sub-tasks. LCP–2021 [147] is described further within Section 6.4.

4.6 Numerical Complexity

Complexity prediction research has also included the identification and simplification of complex numerical expressions. Complex numerical expressions refer to "dates, measurements, quantities, percentages, or ratios" [20], that children, individuals with poor numeracy, or a learning disability may find to be difficult to interpret [17, 19, 20]. These numerical expressions can be presented either numerically, for instance, "*25%*," "*>25*," or "$\frac{1}{4}$," or lexically, as is the case for "*twenty five percent*," "*greater than 25*," or "*a quarter*" [20]. The purpose of numerical complexity prediction is to identify which numerical expressions are considered complex and are therefore in need of simplification for a specific target demographic.

Rello et al. [132] conducted an eye-tracking study to gauge the cognitive load associated with numerical expressions when presented as digits compared to when presented as lexical items. They discovered that digits were easier to read for people with dyslexia than compared to words describing numerical expressions.

Bautista and Saggion [20] have since created a rule-based system for automatically identifying and simplifying complex numerical expressions in Spanish. They handcrafted numerous rules that utilized regular expressions to identify and then simplify complex numerical expressions within 59 sentences. Their system achieved an F1-score of 0.93 on a manually annotated gold-standard dataset and was subsequently considered to have an acceptable level of performance. Bautista et al. [18] later incorporated this system within a more generic TS model.

5 EVALUATION METRICS

The performance of complexity prediction systems is measured using a variety of evaluation metrics. These evaluation metrics depend on the task, with the most common tasks being (1) binary classification performed by prior CWI systems [115, 155, 185], or (2) regression conducted by LCP systems [147], as described within Section 4. The following evaluation metrics were used in the international competitions listed within Section 6.

5.1 Evaluating CWI Systems

The performance of binary CWI systems was normally measured using accuracy, precision, recall, F1-score, and G-score. Accuracy is simply the fraction of positive predictions made over the total number observations within the dataset, precision is "the fraction of positive predictions made that are correct" [70], whereas recall is "the fraction of the truly positive instances that the classifier recognizes" [70].

F1-Score. F1-score is the harmonic average of the accuracy and recall scores [70]. It is subsequently far more informative for evaluating CWI performance as it penalizes those systems that demonstrate either low precision and recall or a high imbalance between the two [70]. Per class F1-scores are then used to calculate macro and weighted F1-scores for all systems. Macro F1-score being the arithmetic mean of all per-class F1-scores, and weighted F1-score being the mean of all per-class F1-scores while taking into consideration the number of actual occurrences of each class within the dataset.[5] F1-score is calculated using the equation below (Equation (4)).

$$F1 = 2\frac{Precision \cdot Recall}{Precision + Recall}. \tag{4}$$

Finally, in CWI 2016 [115], the organizers used G-scores which, unlike F1-score, takes into account accuracy and recall rather than precision and recall.

[5] See Hackeling [70] for further details regarding macro and weighted F1-scores as well as how to calculate accuracy, precision, and recall.

5.2 Evaluating LCP Systems

Recent LCP systems designed to predict continuous instead of binary complexity values are commonly evaluated using mean absolute error, mean squared error, Pearson Correlation, and Spearman's Rank.

Mean Absolute Error. Mean absolute error (MAE) is the average absolute difference between the predicted observations and the actual observations made. It is calculated using the following equation (Equation (5)).

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}, \tag{5}$$

where n is the total number of observations, i is the current observation, y is the predicted observation, and x is the actual observation seen. The closer a MAE value is to zero, the greater the system's performance.

Mean Squared Error. Mean squared error (MSE) is the average squared difference between the predicted observations and the actual observations made. MSE is used to understand the variance and the bias of the predicted observations. Variance refers to the spread of the predicated observations. Bias refers to the spread of the predicted observations compared to that of the actual observations. MSE is produced by the following equation (Equation (6)).

$$MSE = \frac{\sum_{i=1}^{n} (y_i - x_i)^2}{n}, \tag{6}$$

where n is once again the total number of observations, i is the current observation, y is the predicted observation, and x is the actual observation seen. An MSE closer to zero may indicate the presence of less outliers within the provided dataset.

Pearson's Correlation. Pearson's Correlation (R) was the primary means of evaluation in LCP–2021 (See Section 6.4) [147]. It measures the linear relationship between two variables and returns a value between −1 and 1 with a returned value closer to −1 indicating a strong negative correlation, whereas a returned value closer to 1 indicates a strong positive correlation. Pearson's correlation is calculated using the following equation (Equation (7)).

$$R_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}, \tag{7}$$

where X and Y are the variables taken into consideration, σ is the standard deviation, and *cov(X,Y)* stands for co-variance of the two variables.

Spearman's Rank. Spearman's Rank (ρ) takes into consideration the monotonic relationship between two variables, even if this relationship is not linear. Therefore, unlike Pearson's Correlation, Spearman's Rank is more robust when dealing with outliers. It also returns a value between −1 and 1 that depicts the same associated correlations: strong negative (−1) and strong positive (1). Spearman's Rank is provided by the following equation (Equation (8)).

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}, \tag{8}$$

where d_i is the difference between the two ranks of each observation, and n is the total number of observations.

6 INTERNATIONAL COMPETITIONS

LCP has been the focus of several international competitions, known as shared tasks. These shared tasks have been described throughout the following sections (Sections 6.1–6.4). Further details regarding the architecture, development, and evolution of the systems submitted to these shared tasks has been provided in Section 7. In addition, system summaries have been provided in the Appendices in Tables A.1–A.3.

6.1 CWI–2016 at SemEval

The first CWI shared task, referred to as CWI–2016, was organized at the **International Workshop on Semantic Evaluation** (**SemEval**).[6] CWI–2016 (SemEval–2016 Task 11) was modeled as a binary classification task. Participants developed systems to predict the complexity value of English words in context. The organizers provided a dataset sampled from various sources such as the CW Corpus [142], the LexMTurk Corpus [74], and Simple Wikipedia [45].

The target words in the CWI–2016 dataset were annotated by a pool of 400 non-native English speaking annotators. The CWI–2016 dataset was split into a training and a test set. The training set included 2,237 target words in 200 sentences each annotated by 20 annotators. A word was considered complex in the training set if at least one of the 20 annotators assigned it as such. The test set included 88,221 target words in 9,000 sentences each annotated by a single annotator. According to the organizers of CWI–2016, this setup was devised to imitate a scenario where the goal was to predict the individual needs of a speaker based on the needs of the target group [115]. Finally, in terms of task setup, CWI–2016 considered only single word annotations while MWEs were not considered.

A total of 21 teams submitted 42 systems to CWI–2016 and 19 of them wrote system description papers published in the SemEval proceedings. Participants used a wide range of models and features summarized in Table A.1 (Appendices) and discussed further in Section 7.

Most teams who participated in the shared task used simple probabilistic models trained on features such as n-grams, word frequency, and word length. The approaches used by the top-three systems in CWI–2016, being PLUJAGH [179], LTG [99], and MAZA [100], also relied on probabilistic classifiers and on the aforementioned features. The F1-scores achieved by the top-three systems were 0.353, 0.312, and 0.308, respectively, which were considered rather low compared to the baselines and the post-competition analysis presented in Zampieri et al. [193]. According to Zampieri et al. [193], this indicated that CWI–2016 was a particularly challenging task due to the data annotation protocol and the training/test split, since 40 times more testing data was available compared to the training data.

6.2 CWI–2018 at BEA

The second edition of the CWI shared task,[7] referred to as CWI–2018, was organized at the Workshop on the Innovative Use of NLP for **Building Educational Applications** (**BEA**). CWI–2018 was a multilingual shared task featuring datasets containing English, French, German, and Spanish data. A total of three tracks were available, namely, English, German, and Spanish monolingual, with a fourth additional track being made available at a later date. Furthermore, training and testing data from the multi-domain *CWIG3G2* dataset [186] was available for each initial language. The fourth track was the French multilingual track where only a French test set was available and the

[6] http://alt.qcri.org/semeval2016/task11/.

[7] https://sites.google.com/view/cwisharedtask2018/.

Table 7. Systems Submitted to the ALexS–2020 in Alphabetical Order as Summarized by [192]

Team	Classifiers	Features	Paper
UDLAP	Threshold-based	General lexicon, specialized lexicon of internet-related terms, n-grams, frequency.	[160]
Vicomtech	Gaussian Mixture Models (GMM) and *K*-Means Clustering	Lemma length, lemma frequency in subject documents, number of synsets in WordNet, lemma frequency in domain corpora, lemma probability in domain corpora, word frequency in Wikipedia, and word probability in Wikipedia.	[197]
HULAT	Support Vector Machine	Word length, a Boolean determining whether only capital letters were used, a Boolean determining a target word's inclusion in an easy-to-read lexicon, Word2Vec vectors, and BERT vectors.	[5]

participants had to use the data made available for the other three languages to make predictions in French (Section 9.2.6).[8]

The CWI–2018 datasets were split on training, development, and testing partitions. The English dataset contained 27,299 instances for training, 3,328 for development, and 4,252 for testing. The Spanish dataset featured 13,750 instances for training, 1,622 for development, and 2,233 for testing. The German dataset included 6,151 for training, 795 for development, and 959 for testing. Finally, the French dataset only included a testing partition with 2,251 instances.

The three main new aspects of CWI–2018 compared to CWI–2016 were (1) its multilingual nature compared to the English-only CWI–2016, (2) the presence of both target single words and multiple consequent words, and (3) two sub-tasks, one modeled as a binary classification task, and one modeled as a probabilistic classification task.

CWI–2018 received submissions by 12 teams in multiple task and track combinations. At the end of the competition, 10 teams wrote system description papers presented at the BEA workshop. In Table A.2 (Appendices), we present the approaches by teams who submitted systems to the CWI–2018 English binary classification task and who also wrote system description papers. An observed trend was that more teams tried deep neural networks in CWI–2018 compared to CWI–2016, a trend also observed in other areas of AI and NLP research (Section 7).

In CWI–2018's binary classification task, being sub-task 1, the organizers reported the performance from all teams in each of the three domains, namely, *News*, *WikiNews*, and *Wikipedia*. As discussed in the CWI–2018 report [185], the performance obtained by all teams on the News domain was generally substantially higher than the performance obtained in the two other domains.

6.3 ALexS–2020 at SEPLN

ALexS–2020 [192], referring to the lexical analysis shared task at the **Intentional Conference of the Spanish Society on Natural Language Processing** (**SEPLN**), was the first shared task to look at CWI for Spanish educational texts.

The shared task included a Spanish dataset consisting of 9,175 words, with 723 of these words being identified by 430 student annotators as complex. These words were taken from transcripts of academic videos in Spanish made within the University of Guayaquil, Ecuador. Teams were challenged with creating a system to automatically identify which of these 9,175 were labeled as complex.

Three teams participated at ALexS–2020. Each team was presented with the entire dataset, with only the total number of complex words being revealed. As such, no development or training partitions were provided. This encouraged the development of several models as shown within Table 7.

The performances achieved at ALexS–2020 were considered to be poor. The best performing system by UDLAP [160] attained a macro F1-score of 0.272, whereas the best performing systems of

[8] See Section 9.2.6 for more information regarding this type of complexity prediction: cross-lingual LCP.

VIcomtech [197] and HULAT [5] achieved macro F1-scores of 0.176 and 0.164, respectively. These low performances indicated the overall difficulty of the task, since not being presented with a training or development set led to the teams having no idea what was considered to be characteristic of a complex word within the particular domain of Spanish educational texts.

6.4 LCP–2021 at SemEval

The 2021 Lexical Complexity Prediction Task [147], referred to as LCP–2021, was also held at SemEval and attracted 58 teams across its two sub-tasks as shown within Table A.3 (Appendices).

The dataset [146] was developed using crowd sourcing. 10,800 instances were selected from three corpora covering the Bible [39], biomedical articles [82], and europarl [16]. LCP–2021's dataset contained single words (9,000 instances) and MWEs (1,800 instances). The MWEs were limited to pairs of nouns, or adjective-noun collocations. The annotated tokens were presented in context to both the original annotators and the participating teams. This meant that the complexity assignments were not only for the token, but instead for the token in its contextual usage. Multiple instances of tokens were included in different contexts, each receiving differing contextual complexity assignments. As such, systems that took context into account fared well in the final evaluation.

The organizers split the dataset into trial, train, and test sets, stratifying the data for the token type, token instance, complexity, and genre. This meant that even distributions of MWEs and single words were available in each subset as well as an even distribution across genres. Complexity labels were also evenly distributed between the subsets with each having a similar spread of labels. The repeated occurrences of tokens were grouped together in each subset, such that no subset shared any tokens with another subset to prevent information bleed between subsets.

The shared task allowed participants to submit to one of two sub-tasks. The first sub-task permitted systems to only predict the complexity values of the single word instances within the CompLex dataset [146]. The second sub-task asked participants to predict the complexity values for the entire dataset, forcing them to develop a methodology for adapting their single word models to the MWE use case. The organizers did not evaluate solely on MWEs due to the smaller size of the subset. All data was collected via CodaLab and the systems were ranked according to their Pearson's Correlation with the held-back gold standard labels on the test sets.

Several of the top-ranking systems for LCP–2021's sub-task 1 used transformer-based models [167]. However, systems that used handcrafted features [108, 168, 181] also performed well with the top performing system [183] in this category having achieved third place on the official ranking table. This is discussed further within Section 7.4.

Sub-task 2 saw fewer participants than sub-task 1 (37 teams in total). Systems used similar models to those in sub-task 1, with the key difference being the strategy for combining MWEs. Feature-based systems were able to average the features [118, 150, 181] or predictions [108] for each token in an MWE to give the overall value. Deep learning–based systems were typically able to encode the MWE as part of their existing training scheme by supplying the transformer architecture with two encoded tokens instead of one.

7 APPROACHES TO PREDICTING LEXICAL COMPLEXITY IN ENGLISH TEXTS

Various ML models have been used for LCP. These range from SVMs, **decision trees** (**DTs**), random forests (RFs), neural networks to state-of-the-art transformers, such as BERT [52], RoBERTa [94], and ELECTRA [40]. Many of these models have also been used in unison to form ensemble-based models. Prior to more recent transformer-based models, ensemble-based models that utilized multiple DTs, RFs, or neural networks, were state-of-the-art in predicting lexical complexity [115, 185]. This section describes in detail the various models used for LCP. It demonstrates the evolution of LCP systems by providing their model's architecture and performance.

7.1 Machine Learning Classifiers

7.1.1 Support Vector Machines. SVMs are statistical classifiers. They use labeled training data and engineered features to predict the class of unseen inputs [44, 142]. SVMs are well suited for binary classification. They achieve exceptional performance when there exists a clear distinction between two classes. SVMs work less well when dealing with multiple classes or a large number of features as this reduces the uniqueness of each class. SVMs were popular within early LCP research which focused on binary complexity prediction [77, 155].

Jauhar and Specia [77] were one of the first to adopt an SVM for complexity prediction. They trained their SVM on three types of features: morphological, contextual, and psycholinguistic. Morphological features were generated through the use of character n-grams. Contextual features were obtained through a bag-of-words approach, whereby n-grams were used to select neighboring words. Psycholinguistic features were in relation to a target word's degree of concreteness, imageability, familiarity, and age-of-acquisition. Their SVM outperformed a prior baseline CWI model trained on word frequencies.

Shardlow [142] created a complex word corpus (the CW Corpus) consisting of 731 complex words in context [143] (Section 9.1.1). He then experimented with a variety of simplification techniques, including the use of a SVM for binary complexity prediction. His SVM was trained on several features. These features being word frequency, syllable count, word senses, and synonyms associated with the target word. His SVM achieved a higher recall over its precision. This indicated that his SVM was good at identifying complex words, yet often missclassified non-complex words as being complex. It was subsequently prone to the word boundary misclassification problem that is associated with binary CWI systems (Section 4.2.1).

Kuru [86] was interested in the use of Glove word embeddings [124] for capturing the contextual information of a target word. Building on Jauhar and Specia [77]'s bag-of-words approach in extracting contextual information, Kuru [86] investigated how effective Glove word embeddings, or vectors representations, were at CWI when used as features. They trained two SVM models, referred to as AIKU native and AIKU native1, which they submitted to CWI–2016 [115]. The first model, AIKU native, was trained on the "word embedding of the target word and its substrings as features" [86]. The second model, AIKU native1, was trained on the word embedding of the target word, its substrings, as well as the embeddings of the target word's neighboring words. They discovered that both of their models performed equally well having attained matching G-scores of 0.545 at CWI–2016 [115]. This led Kuru [86] to conclude that contextual information, such as a target word's neighboring words, was not a useful feature in improving the CWI performance of a SVM model.

Sanjay et al. [140] experimented with Word2vec word embeddings alongside statistical, POS tag, and similarity features. They trained four SVM models. Their first model was trained on Word2vec word embeddings. Their second model was trained on Word2vec word embeddings, word length, number of syllables, ambiguity count, and frequency. Their third model was trained on Word2vec word embeddings and the similarities between the target word and its neighboring words. Their fourth model was trained on all of the above features, taking into consideration word embeddings, along with statistical and contextual features. The fourth model was found to be the best. Submitted as AmritaCEN (w2vecSim) to CWI–2016, it achieved a F1-score of 0.109 and a G-score of 0.547 [115, 140]. It comes as no surprise that given their reliance on word embeddings and contextual information, Sanjay et al. [140]'s AmritaCEN (w2vecSim) and Kuru [86]'s AIKU (native1) have both achieved similar performances. However, an interesting observation is that Sanjay et al. [140]'s fourth model with the addition of POS tags: AmritaCEN (w2vecSimPos), performed less well. This would suggest that POS tags are less important for CWI than previously theorized. This is supported by the performance of POS tags as a feature for LCP as shown by Desai et al. [51].

7.1.2 Decision Trees. DTs make predictions based on a set of learned sequential or hierarchical rules housed in decision nodes, or leafs. They apply a top-down approach, filtering labeled data through various decision nodes, or branches, until that data is separated as accurately as possible in accordance to class. As such, DTs are often found to surpass the performance of SVMs at LCP [115]. This may be due to DTs being better suited in dealing with features that overlap between classes, given their reliance on learned rules rather than prototypical features, such as support vectors.

Throughout CWI–2016, as detailed in Section 6.1, the most common and arguably the most successful CWI systems consisted of either a DT or a RF model [115]. This marked LCP's transition to DTs and RFs. These models maintained state-of-the-art status until LCP–2021 [147]. This is partly due to these models being trained on a greater number of varied and unique features related to lexical complexity. The use of these additional features was inspired by Shardlow [142], Jauhar and Specia [77], and others' success at surpassing previous baseline performances. It is also partly due to the use of DTs and RFs within ensemble-base models; this is described in greater detail within Section 7.2.

Choubey and Pateria [38] investigated the performance of both a SVM and a DT at CWI. They discovered that their "SVM seemed to be less effective for CWI" [38, 115]. Their SVM attained an F1-score of 0.179 and a G-score of 0.508, whereas their DT produced an F1-score of 0.181 and a G-score of 0.529 [38]. They reasoned that their SVM's slightly worst performance was due to it having "overlapping decision boundaries" [38]. Again, this refers to the decision boundary misclassification problem that is commonly faced by CWI systems (Section 4.2.1).

The systems submitted by Quijada and Medero [129], referred to as team HMC, were among the top performing systems at CWI–2016 [115, 129]. One of HMC's systems consisted of a DT, known as HMC-DecisionTree25, whereas the another consisted of a **regression tree** (**RT**), named HMC-RegressionTree05. These models outperformed their SVM counterpart, with the DT model achieving an F1-score of 0.298 and a G-score of 0.765. Both models were set to have a maximum depth of three, meaning that only three decision nodes, or rules, were learned. These rules were learned from several inputted features. These features belonged to two main categories: statistical and psycholinguistic.[9] Their statistical features included unigram and lemma frequencies; word, stem, and lemma length; probability of a word's character sequence; and lastly, number of synsets, whereas their psycholinguistic features included age-of-acquisition, perceived word concreteness, and the number of differing pronunciations associated with a target word [129]. They claimed that their models' success was due to their use of corpus-based features, especially their use of unigram and lemma frequencies.

7.1.3 Random Forests. RFs consist of multiple DTs. Each DT is trained on a random subset of the training data. From their limited input, each DT then learns a sequence of hierarchical rules for classification. A RF's final output is generated through a plurality voting system. Since each DT only observes a small fraction of the training data, it results in RFs being less prone to overfitting. Each DT learns to distinguish its inputted classes without making sweeping generalizations across the entire dataset. This means that each DT becomes specialized at identifying the distinguishing features of its limited input. Pooling these DTs together subsequently makes for a RF that is more adaptable to unseen data than a stand-alone DT. A RF is, therefore, better suited at dealing with a large dataset with a large number of features compared to a single DT.

[9] HMC also utilized POS tags as features.

Ronzano et al. [136] submitted a RF to CWI–2016 that outperformed other DT models [115]. Their RF, referred to as TALN (RandomForest_WEI), was taken from the Weka ML framework [71]. Being a RF, it consisted of several DTs trained on multiple features, many of which being similar to the features used by the two HMC systems [129]. However, like other models submitted to CWI–2016, additional features were also exploited, such as contextual features [115]. These contextual features took into consideration the position of the target word within a sentence, the number of tokens within that sentence, and the frequencies of both the target word and its context words within the **British National Corpus** (**BNC**) [42, 90] and the 2014 English Wikipedia Corpus [43].[10] The use of such contextual features, together with its RF architecture, may explain TALN's superior performance in comparison to HMC's DT and RT models [129]. TALN (RandomForest_WEI) achieved an F1-score of 0.268 and a G-score of 0.772. This was, respectively, −0.02 less than the F1-score and +0.006 better than the G-score achieved by the best performing HMC system [115, 136].

Zampieri et al. [194] created a CWI system, referred to as MACSAAR (RFC), with a particular focus on Zipfian features. Zipf's Law implies that words that appear less frequently within a text are longer and as a result are likely to be considered more complex than words that are more frequent and shorter [129, 194]. To test this assumption, they trained a SVM, RF, and **nearest neighbor classifier** (**NNC**) using a variety of Zipfian features. These features included word frequency, word and sentence length, and the sum probabilities of the character trigrams belonging to the target word or to the sentence. Their RF model was their best performing model. It attained an F1-score of 0.270 and a G-score of 0.754 at CWI–2016 [115] giving it a greater F1-score of +0.002, yet an inferior G-score of −0.018 compared to TALN [136]. Per their model's performance, Zampieri et al. [194] concluded that Zipfian features are good baseline indicators of lexical complexity.

Davoodi and Kosseim [48] experimented with several models for CWI–2016 [115]. These models were a naïve Bayes, a neural network, a DT, and a RF. Their best performing model was their RF, referred to as CLacEDLK (CLacEDLK-RF_0.6). This model was trained on several features. Davoodi and Kosseim [48] had a particular interest in psycholinguistic features, namely, abstractness. They believed there existed a correlation between "the degree of abstractness of a word and its perceived complexity"[11] [48]. They developed two RF models. Their first RF had a threshold of 0.5, whereas their second RF had a threshold of 0.6. This meant that for a target word to be classified as being complex, these RFs' sub-DTs' output would have on average a complexity value above 0.5 for their first RF and above 0.6 for their second RF. Their second RF was found to outperform their first by a G-score of +0.028. As such, having a higher threshold for complexity assignment would appear to improve CWI performance.

7.1.4 What are the Best Linguistic Features for Predicting Lexical Complexity? The SVMs, DTs, and RFs described above have all so far utilized a common set of features that can be separated into four categories: statistical, morpho-syntactic, psycholinguistics, and contextual. Work by Desai et al. [51], Tack [161], as well as Shardlow et al. [148] have since demonstrated that such statistical features, such as word length, word frequency, and syllable count, psycholinguistic features, including prevalence (average familiarity), age-of-acquisition, and concreteness, together with contextual features, the likes of character or word-level n-grams, continue to be good predictors of lexical complexity.

[10] The presence of low- or high-frequency context words was believed to be an indicator of a target word's degree of complexity. If on average, a target word was surrounded by more highly frequent context words, then that target word was believed to be non-complex, whereas if it were surrounded by less frequent words, then that target word was believed to be complex.

[11] Non-complex words are theorized to have more concrete meanings than complex words, hence complex words are believed to be more abstract in regard to their meaning [32].

Recently, Desai et al. [51] went as far as to rank the effectiveness of several features using a RF trained on the CompLex dataset [146] (Section 6.4). They discovered that prevalence, age-of-acquisition, and concreteness achieved the first, second, and third best performances, respectively, and POS tags and prior complexity labels achieved the worst performances. However, apart from the use of character-level bigrams, Desai et al. [51] failed to investigate the effect contextual features would have had on their model's performance. Contextual features have also been exploited in ensemble-based models, neural networks, and state-of-the-art transformers. The impact of these models' use of contextual features is discussed in Section 7.4.1.

7.2 Ensemble-Based Models

A RF is an ensemble-based model. An ensemble-based model is any model that is made up of multiple sub-models and that produces a final output through some form of plurality voting. These sub-models can be of the same type, as is the case for an RF, or of differing types. The main advantages of ensemble-based models are brought about through their diversity. An ensemble-base model can utilize the strengths of various models, be it either SVMs, DTs, RFs, neural networks, or even transformers, while simultaneously mitigating the disadvantages associated with using only one type of model. As a consequence, ensemble-based models are state-of-the-art for LCP. However, throughout the years, differing combinations of sub-models have been used. From CWI–2016 [115] to CWI–2018 [185], the best performing ensemble-based models consisted of a combination of DTs, RFs, or neural networks. Since LCP–2021, this has changed. State-of-the-art ensemble-based models now consist of various transformers (Section 7.3.1).

Malmasi and Zampieri [100] built upon the use of multiple DTs, hence a RF for binary CWI. They adopted a meta-classifier architecture. A meta-classifier architecture is a unique type of ensemble-based model. It "is generally composed of an ensemble of base classifiers that each make predictions for all of the inputted data" [100]. These base classifiers then input their output into a second set of classifiers. This second set of classifiers, or meta-classifiers, take as features the output of the first set of base classifiers. They then produce their own output through "a plurality voting process" [100].

Malmasi and Zampieri [100] submitted two ensemble-based models to CWI–2016: MAZA A and MAZA B [115]. Both of these models' base classifiers were decision stumps, which are different from DTs as they are trained on a single feature and subsequently only have one decision node, thus giving them the appearance of a tree stump rather than of an entire tree. Bootstrap aggregation was then applied to the output of each decision stump. This bagged output was then inputted into a second level of meta-classifiers consisting of "200 bagged decision trees" [100].

MAZA B was trained using additional contextual features that were not utilized by MAZA A [100]. These contextual features were also different from those used by other aforementioned systems. Together with word frequencies, MAZA B also incorporated two types of probability scores as contextual features. The first being conditional probabilities, being the probability of a target word appearing next to its neighboring one or two words. The second being joint probabilities, being the probability of a target word occurring in conjunction with its surrounding words within a sentence. As such, MAZA B was found to outperform MAZA A. It achieved an F1-score of +0.116 greater than MAZA A [100, 115]. Malmasi and Zampieri [100] attributed this superior performance to MAZA B's use of contextual features, highlighting the importance to which they believed context influences a word's perceived level of complexity.[12]

[12] Malmasi and Zampieri [100] would appear to contradict Kuru [86], as Kuru [86] found context to be uninfluential on his SVM's performance. Early LCP research debated the importance of context. However, context is now more firmly believed to be an influential factor within current LCP literature [112, 183] (see Section 7.4.1 for further details).

Choubey and Pateria [38] constructed two ensemble-based models for CWI–2016 [115]. The first, referred to as GARUDA (HSVM&DT), had a meta-classifier architecture comprised of five SVMs and five DTs. In this model, the SVMs were the base classifiers tasked with the binary classification task of CWI. Its second set of meta-classifiers were its DTs. These meta-classifiers identified whether the predictions made by its SVMs were correct or incorrect. Choubey and Pateria [38]'s second ensemble-based model contained 20 SVMs. Unlike their first model, their second model did not employ meta-classifiers. Instead, each of the 20 SVMs were tasked with predicting the labels of the entire training set. The best performing SVMs then had the most impact in calculating the model's final output labels through a performance-oriented voting system. Interestingly, their first ensemble-based model was found to perform worse than individual SVM or DT models, whereas their second ensemble-based model achieved average performance. They blamed this poor performance on the "overlapping decision boundaries" [38] of their SVM sub-models. This once again demonstrates the inferiority of SVMs for CWI compared to other models.

The SV000gg systems, created by Paetzold and Specia [115], were the best performing systems submitted to CWI–2016 [115, 116]. Paetzold and Specia [115] adopted ensemble-based models that utilized a variety of sub-models. They believed that model diversity would result in greater CWI performance. They experimented with ensemble-based models that consisted of a lexicon-based model, a threshold-based model to SVMs, DTs, RFs, and other ML classifiers. Their lexicon-based model identified whether a target word was a complex or a non-complex word by searching for that word within a given dictionary of pre-labeled lexemes. Their threshold-based model separated complex and non-complex words by seeing whether a target word had a particular feature above a certain threshold and that was also found to be a defining characteristic of that word type (see Section 8.2 for more information regarding lexicon-based and threshold-based approaches to predicting lexical complexity).

The predictions made by their diverse set of sub-models were counted and then used to determine the system's final output through hard or soft voting. As such, there were two versions of the SV000gg system: Hard SV000gg and Soft SV000gg. Hard SV000gg used hard voting to produce the final output label by counting how many times in total the target word was labeled as being either complex or non-complex by all of its contained sub-models. Soft SV000gg used a form of performance-oriented soft voting. Traditional soft voting generates a summed confidence estimate in regard to how likely a target word belongs to a particular class. The final label assigned to this word is then resulted from this summed confidence estimate. Performance-oriented soft voting determines the final label of a target word by examining the performances of each sub-model "over a certain validation set such as precision, recall, and accuracy" [116]. The most common label produced by these sub-models with the highest overall performance, is then chosen as the final output label.

Soft SV000gg achieved the best performance with an F1-score of 0.246 and a G-score of 0.774. Hard SV000gg attained a slightly worse F1-score and G-score of 0.235 and 0.773, respectively. However, Hard SV000gg still outperformed all of the other systems submitted to CWI–2016 in regard to its G-score, including those mentioned above [115]. As a result, both models demonstrated the superiority of diverse ensemble-based models for binary CWI in comparison to other models.

Gooding and Kochmar [64] were inspired by the performance of prior ensemble-based models at CWI–2016 [115]. Their system, referred to as Camb, ranked first on both of CWI–2018's sub-tasks, binary CWI and probabilistic complexity prediction, when dealing with English monolingual data (Section 6) [185]. Camb used a boosting classifier, AdaBoost, with 5,000 estimators followed by a RF bootstrap aggregation model [64]. They experimented with differing sub-models, each being trained on a set of given features similar to those used by prior CWI systems [115]. They concluded that an ensemble-based model that combines both AdaBoost and a RF with equal weights, consistently produced the best performance [64].

Aroyehun et al. [14] experimented with the tree learner model provided by KNIME [21], along with other combinations of DTs, RFs, and gradient boosted tree learners for CWI–2018's sub-task 2: probabilistic complexity prediction [14, 115]. They found that their KNIME tree learner model obtained good results when set to contain 600 models. It achieved a mean macro F1-score of 0.818 across the three datasets provided by CWI–2018 (Section 6). Therefore, Gooding and Kochmar [64] and Aroyehun et al. [14] have demonstrated that ensemble-based models achieve good performance at binary as well as probabilistic complexity prediction.

7.3 Neural Networks

Deep learning is highly popular within NLP and Computational Linguistics having achieved state-of-the-art performance in various NLP-related tasks [54, 180]. Neural networks attempt to mimic human learning by artificially replicating the neuroplasticity of the human brain. They achieve this by manipulating weight values (synaptic strength) between nodes (neurons) that contain characteristic information, or learned features, related to the input (or environmental experience as is the case with the human brain). These weight values are adjusted through a loss function applied after each epoch, or iteration. This process is repeated until these weight values are fully optimized and the most optimum output is produced.

Neural networks can be either supervised or unsupervised. This means that they can learn such characteristic information, or features associated with a complex word, independently. However, within LCP research, neural networks have consistently under-performed in comparison to other more traditional feature engineered models, such as DTs or RFs. This is especially true when such traditional models have been combined within ensemble-based models [115, 185]. It was not until the introduction of continuous complexity prediction in the form of probabilistic complexity (Section 4.3), that some neural networks were shown to perform well, and on occasion, on par with more traditional models [14, 185].

Gillin [60] was one of the first to investigate the performance of a recurrent neural network (RNN) at binary CWI. Within their RNN, they included a **gated recurrent unit** (**GRU**). A GRU is designed to safeguard against the vanishing gradient problem. The vanishing gradient problem arises during back-propagation, when the neural network adjusts its loss function in accordance to its current prediction. The vanishing gradient problem refers to when the gradient of the loss becomes excessively small over time, thus inhibiting the weight values of earlier nodes from being accurately updated [60, 61]. This impairs a neural network's ability to retain information learned at earlier stages. A GRU counters this problem by acting as a "memory" device [60]. It controls what new information should be learned, what prior information should be remembered, and what previous information should be forgotten, when updating a weight value.

Gillin [60] created a RNN model with a GRU as well as an ensemble-based model with a meta-classifier architecture. Referred to as Sensible (Combined), their ensemble-based model was built up of five RNNs as base classifiers and a single RF as a meta-classifier. Out of all of the neural network models submitted to CWI–2016, their RNN model with a GRU, referred to as Sensible (baseline), achieved the best performance [60, 115]. Nevertheless, in comparison to other more traditional models, Sensible (baseline) performed poorly. It attained an F1-score of 0.140 and a G-score of 0.646. Gillin [60] claimed it was the small size of CWI-2016's training set that caused their RNN model to perform less well than expected (Section 6.1).

Aroyehun et al. [14] were the first to experiment with a **convolutional neural network** (**CNN**) for binary CWI. A CNN is different from a RNN. It contains an additional convolutional layer that takes as input the output of its first layer and then transforms said input before passing it onto a further layer. However, CNN models lack the temporal capabilities of a RNN with an embedded GRU. Regardless of this limitation, the CNN introduced by Aroyehun et al. [14], referred to as NLP-CIC-CNN, slightly outperformed their ensemble-based model, consisting of various KNIME tree learners, on one out of the three datasets provided by CWI–2018 [185] (Section 7.2). It attained a macro F1-score of 0.855 and an accuracy rating 0.863. This surpassed the macro F1-score and accuracy achieved by their ensemble-based model by +0.003 and +0.004, respectively.

Hartmann and dos Santos [73] compared models that adopted feature engineering to neural networks at CWI–2018 [185]. They trained a variety of models, such as DTs, Gradient Boosting, Extra Trees, AdaBoost, and XGBoost methods, on numerous features including statistical features, such as word length, number of syllables, numbers of senses, hypernyms and hyponyms, along with n-gram log probabilities; again, being similar to those features previously used by prior CWI systems (see Tables A.1 and A.2). These models were compared to a shallow neural network that used word embeddings, and a **Long Short-Term Memory** (**LSTM**) language model capable of handling the vanishing gradient problem through its use of a forget gate along with an additive gradient structure; being parallel to the use of a GRU.

For binary CWI [185], Hartmann and dos Santos [73]'s feature engineered XGBoost model outperformed their neural network models. It attained an F1-score of 0.8606, whereas their shallow neural network and LSTM models achieved F1-scores of 0.8467 and 0.8173, respectively. Nevertheless, for CWI–2018's second sub-task of probabilistic complexity prediction, their LSTM model, referred to as NILC, was superior to all of the other models, having achieved an F1-score of 0.588. Their feature engineered XGBoost model, and their shallow neural network model, achieved less impressive F1-scores of 0.2978 and 0.2958, respectively. Both Aroyehun et al. [14] and Hartmann and dos Santos [73], therefore, proved the viability of using neural networks for probabilistic complexity prediction.

7.3.1 Transformers. The best performing systems of LCP–2021 [147] used transformer-based models. Transformer-based models were introduced to overcome the limitations associated with prior neural networks, such as RNNs, and LSTM models [14, 60, 73]. Vaswani et al. [167] outlines several advantages of transformers, namely, their self-attention mechanism and their ability to more effectively capture long-term dependencies.

Just Blue by Yaseen et al. [183], achieved the highest Pearson's Correlation at LCP–2021's subtask 1 of 0.7886 [147]. It was inspired by the prior state-of-the-art performance of ensemble-based models together with the recent headway in various NLP-related tasks made by transformers [183].

Just Blue consisted of an ensemble of BERT [52] and RoBERTa [94] transformers. This system contained two BERT models as well as two RoBERTa models. Bert1 and RoBERTa1 were fed target words, whereas Bert2 and RoBERTa2 were fed the target words' corresponding sentences, hence context. These models then predicted the lexical complexities of their inputted target words or sentences, whereby their outputted complexity values were determined by weighted averaging. Models 1 had a weight of 80% and models 2 had a weight of 20%. This meant that the complexity of target words was considered to be more important than the complexity of their surrounding words. However, each sentence was still taken into consideration when calculating the weighted average, as prior studies have shown context to be an influential factor on continuous complexity prediction [100, 129]. Once a weighted average was returned by either set of models, BERT and RoBERTa, Just Blue's final output was produced as a simple average of these returned weighted averages.

Yaseen et al. [183] experimented with different models as well as different weight splits between their target word and sentence level inputs. They discovered that between SVM, RF, BERT, and RoBERTa models, along with a BERT and RoBERTa hybrid model, a BERT and RoBERTa hybrid model achieved the highest performance. They also found that between a 90/10, 80/20, and a 70/30 split between target word and sentence level input, a 80/20 weight split, being in favor of the target word, produced the most accurate complexity values. As such, Just Blue's success is likely a result of its diverse ensemble of varying models, as well as its use of, but not over-reliance on, a target word's context.

DeepBlueAI developed by Pan et al. [123] achieved second place at LCP–2021's sub-task 1 and first place at sub-task 2 [123, 147]. It attained a Pearson's Correlation of 0.7882 for sub-task 1 and a Pearson's Correlation of 0.8612 for sub-task 2. It used a variety of pre-trained language models, such as the transformers BERT [52], RoBERTa [94], ALBERT [87], and ERNIE [196]. DeepBlueAI was subsequently an ensemble-based model that used model stacking with five layers. All of its aforementioned transformers were utilized within its first layer. Its second layer then adjusted the transformers' hyperparameters. It manipulated dropout, the number of hidden layers, and the loss function. The third layer then conducted sevenfold cross-validation to check for overfitting or selection bias, with the fourth layer then having adopted training strategies, such as data augmentation and pseudo-labeling. Data augmentation is the training strategy of adding new data to a training set by copying and slightly modifying existing data; in this instance, data from CWI–2018 was used, and for sub-task 2, data from sub-task 1 was used after having gone through "synonym replacement, random insertion, random swap, and random deletion" [123, 176]. Pseudo-labeling is the training strategy of predicting labels for unlabeled data and then adding the newly labeled data back into the training set. The fifth layer contained DeepBLueAI's final estimator in the form of a simple linear regression model. This estimator returned the final predicted complexity values ($\hat{y}$) through the following equation (Equation (9)):

$$\hat{y} = \sum_{j=1}^{N} W_j \hat{y}_j, \tag{9}$$

where N is the total number of transformers with different hyperparameters, W_j is the weight of each transformer, and $\hat{y}_j$ is each transformers' predicted complexity value.

Pan et al. [123] attributed their model's good performance in both sub-tasks to its use of multiple transformers and training strategies. With model diversity also being an influential factor in regard to Just Blue's high performance [183], it would appear that current state-of-the-art LCP systems consist of an ensemble of differing transformer-based models.

RG_PA, created by Rao et al. [130], was the second highest performing system at LCP–2021's sub-task 2 having achieved a Pearson's Correlation of 0.8575 [130, 147]. Unlike Just Blue [183] and DeepBlueAI [123], it did not contain an ensemble of diverse transformers. Alternatively, RG_PA consisted of a single RoBERTa attention-based model. It used **Byte-Pair Encoding** (**BPE**) to firstly tokenize all of its inputted sentences. BPE compresses a given sentence so that its most frequent character pairs, or bytes, are replaced with a single character. This shortens the inputted sentence into a sequence of character representations that help to mitigate the out-of-vocabulary problem.[13] Each of their RoBERTa's hidden layers applied token pooling that creates a vector representation of a target word based on the average of all of the token embeddings of that target word found throughout the training set. The attention weight between the target vector and context tokens, i.e., context words, is then calculated and the returned context vector is concatenated with the target vector. The concatenated vector representation of each target word is then used to predict the complexity values of the unseen words within the test set. Its use of BPE together with its use of concatenated context and target word vectors, may explain RG_PA's high performance in sub-task 2, despite it not being an ensemble-based model.

[13] The out-of-vocabulary problem refers to the problem that arises when a model is presented with a word that was not observed within its training set.

7.4 Other State-of-the-Art Models

The third best performing system at LCP–2021's sub-task 1, deviated from the use of transformer-based models [108]. Mosquera [108] approached sub-task 1 from a more traditional feature engineering approach. Much like prior CWI sytems, Mosquera [108] utilized a combination of lexical, contextual, and semantic features (Section 4.2). However, unlike previous CWI systems, these features were extensive with 51 features in total being used to rate lexical complexity. These features included SUBTLEX features, word etymology, and several readability indices. SUBTLEX features are those features that are embedded within film subtitles, such as the number of films whose subtitles depict the word in lowercase, target word frequency per million subtitled words, as well as the percentage of films where the target word appeared within the SUBTLEX-US corpus [31]. Features related to a word's etymology included the number of Greek or Latin affixes that belong to the target word, and readability index features included Flesch score [56], Gunning-Fog index [69], LIX score [12], SMOG index [103], and Dale-Chall index [36].

Mosquera [108] fed his extensive list of features into a **Light Gradient Boosting Machine** (**LGB**) model with minimal optimization. Results showed that the top three most influential features on LCP performance were age, the Dale-Chall index, and the complexity values taken from Maddela and Xu [97]'s word complexity lexicon. Mosquera [108] also observed that several sentence readability features were top contributors with the Dale-Chall index being the most influential. The Dale-Chall index is a readability index that measures a text's perceived comprehension difficulty by assessing its words familiarity based on a list of 3,000 common words [36]. It is likely that his extensive list of features, the inclusion of such contextual features, or contextual readability measures, along with his use of a LGB model, is responsible for Mosquera [108] outperforming a similar feature engineering approach by Desai et al. [51].

7.4.1 What Effect does the Inclusion of Context have on Predicting Lexical Complexity? Several prior LCP studies that have not yet been mentioned have demonstrated mixed results when it comes to the inclusion of context. On the one hand, Alfter and Pilán [6] experimented with three configurations of the dataset provided by CWI–2018 [185]: "context-free, context-only, and context-sensitive" [6] and found no significant difference in their systems performance between the three. Furthermore, Kriz et al. [84] discovered an increase in their CWI system's performance when one neighboring word was taken into consideration, yet a decrease in its performance when this was increased to two or more neighboring words. On the other hand, Gooding and Kochmar [65] improved their winning system at CWI-2018 [185], SV000gg, by capturing contextual information as a result of converting their model to a sequence labeling task and by using word embeddings. In addition, several studies throughout Sections 7.1.1–7.3, have claimed that the inclusion or inference of contextual features is responsible for their systems high performance [86, 100, 140]. The transformer-based models outlined in Section 7.3.1 also make no exception to this claim and due to their superiority over prior neural networks, such as their ability to more effectively capture long-term dependencies, their exploitation of contextual information is likely even more beneficial [183, 183]. In the past, the effect context had on LCP performance was therefore somewhat debated. However, the high performance of recent models that have included sentence and word-level features, now more definitely suggest that the inclusion of context improves LCP.

7.5 Summary

Current state-of-the-art LCP systems consist of an ensemble of varying transformers. These systems achieve state-of-the-art performance largely due to two reasons: (1) Yaseen et al. [183] and Pan et al. [123] have demonstrated the importance of model diversity within an ensemble-based model, and (2) transformer-based models are better equipped to handle contextual information (Section 7.3). As such, ensemble-based models that rely on multiple transformers of varying types and that take into consideration contextual information of the target word, are currently the state-of-the-art systems for LCP. However, Mosquera [108] has proven that feature engineering is still a viable approach for LCP, given that an extensive set of lexical, contextual, and semantic features are taken into consideration.

8 USE CASES AND APPLICATIONS

LCP has many potential use cases and applications [115, 147, 185]. LCP systems can be utilized within a variety of assistive technologies, such as **computer-assisted language learning** (**CALL**) applications or **intelligent tutoring systems** (**ITSs**) to improve the readability of given texts (Section 8.1). This is most often achieved by implementing **text simplification** (**TS**) that benefits from a LCP component.

8.1 Improving Readability

CALL is the use of any computer-related technology, be it either a word processing document, social media, or other online medium, for language learning. ITSs are "computer learning environments designed to help students master difficult knowledge and skills" [68]. CALL applications subsequently include ITSs that specialize in language learning and have been found to improve second language (L2) acquisition [166]. These applications include multiple designs, are based on differing pedagogical practices, and allow for varying degrees of learner-computer interaction [8].

A common approach among CALL is to simplify a text to make it more accessible for the L2 learner [133, 191]. Alhawiti [7] states that TS can be beneficial to language learners and therefore, an ITS or CALL application that incorporated TS would likewise be beneficial. This is since TS has been found to increase the literacy [125] as well as advance the vocabulary development of L2 learners [133, 162].

Rets and Rogaten [133] tested 37 participants on their ability to memorize and process the ideas presented within two texts: (1) an authentic text, and (2) a simplified text with less complex vocabulary and syntax. Memory was measured by asking the participants to rewrite the observed texts, whereas text processing was gauged through the use of eye tracking. Participants were found to achieve greater memorization and were shown to fixate less on the simplified text than compared to the authentic text. This led Rets and Rogaten [133] to conclude that TS results in better textual comprehension which correlates with a greater learning potential [128, 133].

ITSs that use TS are not restricted to aiding L2 learners. TS improves the readability of texts and thus enhances the literacy development of other target demographics. TS may help an ITS designed for people diagnosed with autism "by reducing the amount of figurative expressions in a text" [151]. It may also increase the effectiveness of ITSs created for people with dyslexia or aphasia. This is by replacing long words with short words, or substituting words with challenging character combinations for those which are easier to identify [35, 131]. ITSs developed for children may likewise use TS in order to reduce the amount of high-level jargon, or uncommon words, within a text [49]. TS is, therefore, useful in improving the vocabulary and literacy development of L2 learners [7], people with autism [151], dyslexia [35, 131], or aphasia [35], as well as children [49]. However, Crossley et al. [46] presents arguments for and against the use of simplified texts within L2 classrooms with Gooding [63] pointing out that the usefulness of simplified texts may vary between target demographics and in some instances may be inferior to alternative reading strategies. Despite this, throughout the years TS systems have assessed lexical complexity through a number of ways.

8.2 LCP's Place in the Text Simplification Pipeline

Prior to the LCP systems outlined within Section 7, TS assessed lexical complexity through several approaches: (1) a simplify everything approach, (2) a threshold-based approach, and (3) a lexicon-based approach [120]. However, each of these approaches had their limitations that led to the development of more dynamic LCP systems.

8.2.1 Simplify Everything.

The simplify everything approach simplified all of the words within a given text [53]. This approach subsequently had no means of identifying complex words. Instead, systems that adopted this approach often used a form of comparative complexity prediction to compare and find the most suitable word replacements for every single word within a provided text. A disadvantage to this approach is that not all words are in need of simplification [144]. As such, systems that adopted this approach often simplified already easy to understand words into equally easy to understand alternatives that were not as well suited as the original word for that particular context [53, 119]. The simplify everything approach was therefore found to produce ungrammatical and nonsensical simplifications.

8.2.2 Threshold-Based.

Threshold-based approaches required the presence of a feature over a set value in order for a target word to be identified as complex. Systems that adopted this approach often used a single feature-threshold, such as having x number of characters, or x frequency in a certain corpus, as a means of gauging the complexity of a target word [120]. However, this approach was found to be insufficient in identifying all instances of complex words within a given text. For instance, Bott et al. [27] and Shardlow [145] discovered that using word length as a standalone feature-threshold failed to identify complex words which were uncharacteristically short, while incorrectly classifying simple words that were over five characters long. An example being incorrectly classifying *folly* as non-complex yet *foolishness* as complex, since the former may be considered a short word and the latter a long word. As such, the reliance of a single, or sometimes multiple, feature-threshold lost popularity as an accurate means of assessing lexical complexity.

8.2.3 Lexicon-Based.

Lexicon-based approaches utilized a predefined list of words as a means of distinguishing between complex and non-complex words within a given text [120]. Systems that adopt lexicon-based approaches are often found to perform well in identifying complex words for their intended target demographic or domain. However, when identifying complex words for individuals outside of their intended target population or domain, lexicon-based approaches perform less well [120]. For example, FACILITA [174] is designed to distinguish between Portuguese complex and non-complex words for Brazilian children using three dictionaries: (1) consisted of frequent words extracted from Brazilian newspapers, (2) contained concrete words, and (3) housed simple words that were believed to be "common to youngsters" [10]. FACILITA is very effective in helping young low literacy readers in Brazil. However, FACILITA may be less helpful for other demographics, such as second language learners, individuals suffering from a reading disability, or older individuals with low literacy. This is because the words used to make FACILITA's predefined dictionaries may not be considered as easy to understand for these demographics as they were for Brazilian children.

8.3 Other Use Cases

LCP can aid other downstream NLP-related tasks, such as machine translation [170] and authorship identification [1, 156], and is also likely to be beneficial to other downstream tasks in the future. Two alternative use cases of LCP are exemplified in the following Sections 8.3.1 and 8.3.2.

Table 8. Example of a Simplified Sentence Shown in Štajner and Popović [170]

	Sentence
Original	A dozen Chinese fishing **boats** had taken **refuge** in a lagoon of Huangyan Island
Simplified	A dozen Chinese fishing **ships** had taken **shelter** in a lagoon of Huangyan Island

Target words of interest are in bold.

8.3.1 Machine Translation. Before TS shifted to improving the readability of texts, its primary focus was to aid **machine translation** (**MT**) [4]. MT is the task of automatically translating a source language into a target language [170]. MT systems are limited by the lack of parallel corpora that contain identical texts in more than one language. MT systems are also hindered by the morpho-syntactic complexities of the languages that they are tasked to translate. Studies have proven that TS can aid MT [4, 158, 170]. TS achieves this by reducing the ambiguity of the inputted texts in the source language [170]. For instance, by replacing complex words in the source language with simpler alternatives, it increases the probability of an MT system finding a suitable translation in the target language.

Štajner and Popović [170] demonstrated that a TS system that utilized both LS and syntactic simplification components improved the performance of a English-to-Serbian MT system. Their system, being assessed on the adequacy (meaning preservation) and fluency (grammatical correctness) of its output, achieved this by translating simplified sentences rather than translating the original sentences directly.

According to Štajner and Popović [170], the simplified sentence shown in Table 8 resulted in an English-to-Serbian translation that was both easier to understand and more grammatically correct to a group of Serbian annotators than compared to a translation of the original sentence. Without a LCP component, the simplified words, *boats* and *refuge*, may not have been recognized as being complex and as a consequence, would not have been simplified resulting in a less accurate translation. This demonstrates that the inclusion of an LCP component within the TS pipeline can improve MT.

8.3.2 Authorship Identification. Authorship identification is the task of identifying the author of a given text [26]. A text's vocabulary richness is a common feature used for authorship identification. Vocabulary richness is used to capture an individual's linguistic fingerprint, in other words, their idiolect. It is normally measured through the use of the **type-token ratio** (**TTR**). The TTR is "a simple ratio between the number of types and tokens within a text" [85]. The TTR, therefore, shows the diversity of an author's vocabulary. It has been used in such situations as helping to differentiate between authors of highly similar texts [134] as well as to identify the author of online messages [156].

LCP provides an additional measurement of vocabulary richness. Adding to the TTR, it provides an average lexical complexity marker that depicts, on average, how complex the author writes. Average lexical complexity can be inputted into an authorship identification system as a feature that may, in turn, enhance its performance. Tanguy et al. [164] experimented with such a feature, in the form of morphological (lexical) complexity, for the authorship identification of various extracts taken from fictional books. Alternative examples are using lexical complexity to differentiate between authors belonging to different time periods, authors with different levels of education, or authors of different ages; under the assumption that discrepancies exist between their writing styles. For instance, past literature may contain vocabulary considered to be more archaic and complex than modern literature, individuals with a higher level of education may use more jargon-related and complex words than those with a lower level of education, and adults may use more unfamiliar and less common words than children.

9 RESOURCES

9.1 Additional English Datasets and Resources

The shared tasks of CWI–2016 [115], CWI–2018 [185], and LCP–2021 [147], tested participating teams on three datasets that have since contributed significantly to LCP research (Section 6). Nevertheless, these are not the only influential datasets that contain words with lexical complexity ratings. All of the current LCP datasets that deal with English and that are known to the authors' are provided in Table A.4 located within the Appendices. Apart from the CWI–2016, CWI–2018, and the CompLex datasets already discussed within Section 6, the remaining datasets are introduced throughout the following Sections 9.1.1–9.1.4.

9.1.1 CW Corpus. The CW Corpus contains 731 complex words in context [143]. It was constructed using Wikipedia edits. Edits are often made to Wikipedia entries in order to simplify their vocabulary. Using Wikipedia's edit history, it is possible to see the simplified edit as well as the original text. To determine which of these edits contained true lexical simplifications, Shardlow [143] looked at the editor's comments for the word "simple," and calculated Tf-idf vector representations to check for lexical discrepancies between the original and simplified texts. Those texts which were found to contain true lexical simplifications, were then subject to a set of further tests to guarantee the validity of the CW corpus. Hamming distance was calculated to ensure that only one word differed between the original and simplified texts. Reality and inequality checks were conducted to make sure that the target words were known yet different English words, and not just variations of the same word. Lastly, non-synonym pairs were discarded and simplified candidate words were verified. Through these series of checks, 731 complex words were provided with context.

9.1.2 Horn et al. [74]. Horn et al. [74] created a corpus of 25,000 simplified word candidates for comparative complexity prediction. They acquired 50 annotators. These annotators were required to live in the United States in an attempt to control their English proficiency. They were asked to give a simpler alternative for each target complex word within 500 sentences. They achieved this by using Amazon's **Mechanical Turk** (**MTurk**) that is popular among NLP-related tasks [74]. Similar to Shardlow [143], the sentences presented to the annotators were taken from a sentence-aligned Wikipedia corpus. This corpus provided original and simplified Wikipedia entries of the same texts. On average, annotators provided 12 differing simplifications per target word. This makes the corpus introduced by Horn et al. [74] a valuable resource for investigating comparative complexity.

9.1.3 Word Complexity Lexicon. Maddela and Xu [97] recognized the limitations of prior CWI datasets, namely, the limitations associated with using binary complexity labels rather than continuous complexity values (Section 4.2.1) [97]. As a response to these limitations, they constructed the **Word Complexity Lexicon** (**WCL**). The WCL is a dataset made up of "15,000 English words with word complexity values assessed by human annotators" [97]. These 15,000 words were the most frequent 15,000 words found within the Google 1T Ngram Corpus [28]. Their assigned word complexity values were continuous since these values were assigned by 11 non-native yet fluent English speakers using a six-point likert scale. They assigned each word with a value between 1 and 6, with 1 denoting that word as being very simple, and 6 defining that word as being very complex. To determine the final complexity value of each word, complexity values were averaged. Those complexity values which were greater than 2 from the mean of the rest of the ratings, were discarded from the final average. This improved the WCL's inter-annotator agreement to 0.64. The remaining disagreements between the annotators were believed to be due to the differing characteristics of their native languages, hence caused by cross-linguistic influence.

9.1.4 Personalized LS Dataset. Lee and Yeung [89] constructed a dataset of 12,000 words for personalized CWI. These words were ranked on a five-point likert scale. Fifteen learners of English, who were native Japanese speakers, were tasked with rating the complexity of each of the 12,000 words. The five labels that they could choose from ranged between (1) “never seen the word before,” to (5) “absolutely know the word’s meaning” [89]. Lee and Yeung [89] converted these multi-labeled ratings into binary labels. They considered words ranked 1 to 4 as being complex, and words ranked 5 as being non-complex. However, their use of a multi-labeled likert scale means that this dataset can be used for continuous complexity prediction.

The 15 annotators chosen for data annotation were split into two groups of English proficiency. Thus, two subsets of the dataset were created: the low English proficiency subset, and the high English proficiency subset. The low English proficiency subset was annotated by learners whom knew less than 41% of the 12,000 words. The high English proficiency subset was annotated by learners whom knew more than 75% of the 12,000 words. As such, the Personalized LS Dataset [89] is an ideal resource for future personalized LCP research.

9.2 Lexical Complexity Prediction in Languages Other than English

Since CWI–2018 [185] (Section 6.2), LCP for other languages has begun to receive more attention in the form of monolingual, multilingual, and cross-lingual LCP [55, 184]. Monolingual LCP refers to the the task of predicting the complexity values of words in a single language. Multilingual LCP refers to the task of creating a LCP system that can be trained on and used to predict the lexical complexities of multiple languages. Cross-lingual LCP refers to the task of training a LCP system on one or multiple languages and then using that system to predict the lexical complexities of a language previously unseen within the training set.

9.2.1 French, Spanish, and German. As previously mentioned in Section 6.2, the CWI–2018 shared task at BEA [185] contained datasets in French, Spanish, and German. It was discovered that systems generally performed well across these languages, with high performance in one language correlating with high performance in another. The organizers of CWI–2018 saw this as evidence in support of cross-lingual LCP (Section 9.2.6).

Billami et al. [23] was interested in the perceived lexical complexity of French words and as a result created the ReSyf lexicon. This lexicon contains French synonyms that have been ranked in regard to their reading difficulty using a SVM ranker trained on the Manulex resource [96]. Garí Soler et al. [58] investigated the performance of word embeddings at predicting the lexical complexity of French words. They discovered that word embeddings outperformed statistical features, such as word length, number of phonemes, or log frequency when used in isolation. However, when these statistical features were used in unison, they outperformed word embeddings. Other studies interested in French, such as Tack et al. [163] and Tack [161], have already been described in the Personalized Complexity section, Section 4.4.

The ALexS–2020 [192] shared task and its submitted systems [5, 160, 197] sought to predict Spanish lexical complexity and have been introduced within Section 6.3. Merejildo [104] has since detailed that construction of a Spanish CWI corpus. A group of 40 native-speaking Spanish university students were tasked with identifying which words they believed to be complex within 3,887 academic texts. Merejildo [104] conducted feature extraction on the identified complex words and found that word length and frequency were common markers of Spanish lexical complexity.

Apart from several researchers that have participated in CWI–2018 [185] or that have later utilized the CWI–2018 dataset [13, 55], little stand-alone research has been conducted on German LCP.

9.2.2 Chinese. Lee and Yeung [88] created a SVM designed to identify Chinese complex words. Their monolingual LCP model was then further developed by Yeung and Lee [184]. They tasked eight learners of Chinese to rank 600 Chinese words using a five-point likert scale. If the annotator assigned a complexity value of 1 to 3, then that word was labeled as complex. If, however, the word was assigned a complexity value of 4 or 5, then that word was labeled as being either challenging or non-complex, respectively. Their SVM classifier was trained on a number of features parallel to Lee and Yeung [88]; these being, the target word's ranking in a Chinese proficiency test known as the Hanyu Shuiping Kaoshi [72], along with word length, word frequency in the Chinese Wikipedia Corpus [88], and character and word frequency in the Jinan Corpus of Learner Chinese [172]. They discovered that their logistic regression models outperformed their prior SVM [88]. They also found that their model was better at predicting the lexical complexities of their annotators with low Chinese L2 proficiency compared to those with high Chinese L2 proficiency.

9.2.3 Japanese. Nishihara and Kajiwara [111] used a SVM to predict the lexical complexities of Japanese words. They created a new dataset that expanded upon the **Japanese Education Vocabulary** List (**JEV**). JEV contains 18,000 Japanese words divided into three levels of difficulty: easy, medium, or difficult. Nishihara and Kajiwara [111] also rated the complexity of words from Japanese Wikipedia, the Tsukuba Web Corpus [113], and the Corpus of Contemporary Written Japanese [98]. This increased the size of their dataset to 40,605 Japanese words. They trained a monolingual SVM to predict the level of complexity associated with each target word. To achieve this, they used a variety of features that were also used by prior English CWI systems, such as POS tags, character and word frequencies, and word embeddings. However, they discarded other popular features, such as word length, due to the topological and morphological differences between English and Japanese. Unlike English, Japanese "is composed of three types of characters: Hiragana, Katakana, and Kanji" [111]. The characters Hiragana and Katakana are considered simple characters, whereas Kanji are ideographic and are therefore considered more difficult to interpret. As such, in Japanese, the proportion of complex to simple characters within a word is a good indicator of a word's complexity. Nishihara and Kajiwara [111] concluded that the use of such language-specific features was responsible for their model's good performance.

9.2.4 Swedish. Smolenska [153] experimented with a variety of models for binary CWI: SVM, RF, naïve Bayes, gradient boosting, logistic regression, and stochastic descent models. These models were tested on one of two datasets consisting of Swedish words labeled with complexity ratings. The first dataset contained 4,305 Swedish words marked with labels from the **Common European Reference Framework** (**CERF**). These labels ranged from A1, elementary proficiency, to C2, advanced proficiency. The second dataset consisted of 4,238 manually extracted Swedish words from a variety of dictionaries and textbooks that were also labeled with CERF ratings. While evaluating the quality of the two datasets, Smolenska [153] discovered that the second dataset correlated better with the judgments of two human evaluators. Results showed that the RF model achieved the best performance on this dataset having been trained on a number of features, including morpho-syntactic, contextual, conceptual, and frequency-based features.

9.2.5 Multilingual LCP. Sheang [149] saw the advantages of adopting a feature engineering approach as well as a CNN model for multilingual CWI. As a result, Sheang [149] developed a semi-supervised CNN model trained on word embeddings and common CWI features, such as word frequency, word length, syllable and vowel count, term frequency, POS tags, syntactic dependency, and stop words. Being trained on the English, Spanish, and German datasets of CWI–2018 [185] (Section 6.2), this multilingual model was found to outperform the best performing model of CWI–2018 [185] on the Spanish and German datasets.

Aprosio et al. [13] created a LCP system that caters to the native language of the user. As previously discussed within Section 4.4, an annotator's or user's native language influences their perception of lexical complexity through what is known as cross-linguistic influence. As such, their system was designed with the ability to identify the false friends as well as the cognates between the user's native language and the language of the annotated or inputted text. False friends are "those pairs of words in two different languages that are similar in form but semantically divergent" [13]. Cognates, on the other hand, are those pairs of words with the same meaning and similar spelling in two or more languages. Their system firstly identified those words within the inputted text that may be considered cognate. It achieved this by taking into consideration three similarity metrics: XXDICE [29], Normalized Edit Distance [171], and Jaro/Winkler [178]. Once potential cognates had been identified, their system used an SVM to classify which of these cognates may, in fact, be false friends. Their SVM was trained on the cosine similarity between the candidate words, and the cosine similarity between these words' synonyms. Those words which were found to be false friends were labeled as complex, whereas those words which were considered to be cognates and not false friends were labeled as non-complex. Thus, by taking the language of the user into consideration, Aprosio et al. [13] created a LCP system that can recognize and exploit language-dependent features to improve its performance. Aprosio et al. [13] is, therefore, another good example of personalized LCP.

9.2.6 Cross-Lingual LCP. Finnimore et al. [55] continued working on CWI–2018's sub-task 1 [185] (Section 6.2). They focused on the development of a cross-lingual CWI model with a particular focus on discovering which monolingual or multilingual CWI features would also perform well in a cross-lingual setting. They discarded previous features that they believed to be language dependent, hence not transferable from one language to another. They state that the use of word-level n-grams is an example of such a language-dependent feature, since word-level n-grams denote the unique collocations of a particular language. Instead, Finnimore et al. [55] experimented with a variety of features that they believed to be cross-lingual; these features being the number of syllables, tokens, and complex punctuation marks, along with the sentence length and character-level probabilities associated with a target word. They found that training their linear regression model on languages that belonged to the same language family as the target language improved its macro F1-score. However, the inclusion of languages unrelated to that of the target language had the opposite effect. Overall, their cross-lingual model achieved good performance. They go on to state that this is remarkable given its relatively simplistic set of features, thus proving the viability of cross-lingual LCP.

Bingel and Bjerva [24] provide further evidence in favor of cross-lingual LCP. Their cross-lingual CWI system achieved the best F1-score in predicting the lexical complexities of an unseen language, being French. Consistent with other high-performing CWI systems, it was an ensemble-based model that contained "a number of RFs as well as feed-forward neural networks with hard parameter sharing" [24]. Their RFs were trained on a number of features, whereby they discovered that word length and frequency were good cross-lingual predictors of lexical complexity.

Zaharia et al. [189] experimented with several transformer-based models, such as **Multilingual BERT** (**mBERT**) [126] and XLM-RoBERTa [41], for cross-lingual CWI. Both mBERT and XLM-RoBERTa are multilingual masked language models that are pretrained on numerous languages. mBERT is pretrained on "Wikipedia pages of 100 languages with a shared word piece vocabulary" [126]. XLM-RoBERTa is also pretrained on 100 languages, yet with more data [41]. Zaharia et al. [189] tested these models performance on the WikiNews datasets provided by CWI–2018 [185]. They found that XLM-RoBERTa was the best performing model. It achieved a higher F1-score than mBERT on the WikiNews datasets when tasked with predicting the lexical complexities of unseen German or French target words; these F1-scores being +0.02 and +0.04, respectively, greater than that achieved by mBERT. They attributed XLM-RoBERTa's superior performance to its larger pretrained multilingual corpus [41, 189].

9.2.7 Is Transfer Learning Possible for Predicting Lexical Complexity Across Multiple Languages? The studies detailed in Section 9.2.6 provide evidence in favor of transfer learning for cross-lingual LCP. Numerous features, for instance, number of syllables, tokens, complex punctuation marks, and sentence length, have been proven to work well when trained on one language and then used to predict lexical complexities in another [24, 55]. Models, such as mBERT and XLM-RoBERTa, have also been shown to achieve good performances for cross-lingual LCP [189]. With the availability of LCP datasets in high-resource languages (Section 9) and with LCP research gaining traction in languages other than English (Section 9.2), we suspect cross-lingual LCP will become increasingly popular.

10 SUMMARY

This article has presented an overview of LCP research with a specific focus on research conducted on English. It has defined what is meant by "*complexity*" within LCP and has described types of computational modeling applied to its prediction, such as comparative, binary, continuous, and personalized complexity (Sections 2–4). It has provided the evaluation metrics used to evaluate LCP performance and has discussed the international shared tasks that have inspired the creation of numerous LCP systems (Sections 5 and 6): CWI–2016 [115], CWI–2018 [185], and LCP–2021 [147]. It has explained the architecture, development, and evolution of these LCP systems, ranging from feature engineering approaches, neural networks, to the most recent state-of-the-art transformer-based models while discussing relevant research questions within the field (Section 7). It has presented various use cases and applications of LCP, including for other NLP-related tasks such as machine translation (Section 8.3.1) and author identification (Section 8.3.2). It has collected and summarized English datasets (Section 9.1) and has also briefly presented work on languages other than English (Section 9.2).

10.1 Opportunities and Challenges

There now exists an unprecedented demand for LCP research. With distance learning becoming ever more popular and with LCP being a precursor within other NLP-related tasks, the future for LCP research would appear to be promising. LCP–2021 [147] has shown the superiority of transformer-based models for LCP, especially when a diverse set of transformers are used to form an ensemble-based model [123, 183]. CWI–2018 along with others [24, 55, 189], have proven that cross-lingual LCP is viable. LCP is now being conducted for languages other than English [88, 111, 153, 161, 185, 192]. As such, we expect to see ensemble-based models with a diverse set of transformers being used for multi-lingual and cross-lingual LCP. Furthermore, personalized LCP calls for the development of LCP systems with the ability to predict the complexity assignments made by the individual or specific target demographic, rather than belonging to a generalized population [89, 163, 195]. We expect such personalized LCP systems to become popular as their datasets are likely to contain more consistent complexity ratings due to there being less disagreement among their annotators. Research questions investigating such areas as the effect of including context on LCP performance as well as the advantages of complexity prediction of multi-word expressions, are other avenues of LCP research that have likewise proven to aid LCP [67, 183, 183]. We therefore also believe that context and MWEs will continue to be taken into consideration by future LCP systems.

Future LCP research, however, is not without its challenges. A current lack of available data may have already led to some cases of overfitting with models being unable to generalize their predictions across multiple domains or target populations. In addition, dataset quality has previously been put into question, whereby the use of a small pool of annotators, an irregular train/test split, or high levels of inter-annotator disagreement may have led to unreliable complexity labels [193]. To overcome these challenges, we stress the importance of further research into continuous and personalized complexity prediction that takes into consideration context and MWEs, along with the implementation of transfer-learning models for under-resourced languages.

APPENDIX

Table A.1. Systems Submitted to the CWI–2016 in Alphabetical Order as Summarized by [148]

Team	Classifiers	Features	Paper
AI-KU	SVM	word embeddings of the target and surrounding words	[86]
Amrita-CEN	SVM	word embeddings and various semantic and morphological features	[140]
BHASHA	SVM, Decision Tree	lexical and morphological features	[38]
ClacEDLK	Random Forests	semantic, morphological, and psycholinguistic features	[48]
CoastalCPH	Neural Network and Logistic Regression	word frequencies and word embeddings	[25]
HMC	Decision Tree and Random Forest	lexical, semantic, syntactic, and psycholinguistic features	[129]
IIIT	Nearest Centroid	semantic and morphological features	[121]
JUNLP	Random Forest, naïve Bayes	semantic, lexicon-based, morphological, and syntactic features	[110]
LTG	Decision Tree	n-grams and word length	[99]
MACSAAR	Random Forest and SVM	Zipfian frequency distribution, word length	[194]
MAZA	Meta-classifier	n-grams, word probability, word length	[100]
Melbourne	Weighted Random Forests	lexical and semantic features	[30]
PLUJAGH	Threshold-based methods	features extracted from Simple Wikipedia	[179]
Pomona	Threshold-based methods	word frequencies	[80]
Sensible	Ensemble Recurrent Neural Networks	word embeddings	[60]
SV000gg	System voting with threshold	morphological, lexical, and semantic features	[116]
TALN	Random Forest	lexical, morphological, semantic, and syntactic features	[136]
USAAR	Bayesian Ridge classifiers	handcrafted word sense entropy metric and language model perplexity	[102]
UWB	Maximum Entropy	word occurrence counts on Wikipedia documents	[83]

Table A.2. Systems Submitted to the CWI–2018 English Binary Classification Single Word Track in Alphabetical Order as Summarized by [148]

Team	Classifiers	Features	Paper
Camb	Adaboost	n-grams, WordNet features, POS tags, dependency parsing relations, psycholinguistic features	[64]
CFILT_IITB	Voting ensemble	Word length, syllable counts, vowel counts, WordNet-based features	[173]
hu-berlin	naïve Bayes	Character n-grams	[127]
ITEC	LSTM	Word length, word and character embeddings, frequency count, psycholinguistics features	[50]
LaSTUS/TALN	SVM, Random Forest	Word length, word embeddings, semantic and contextual features	[2]
NILC	XGBoost	n-grams, word length, number of syllables, WordNet-based features	[73]
NLP-CIC	Tree Ensembles and CNNs	Word frequency, syntactic and lexical features, psycholinguistic features, and word embeddings	[14]
SB@GU	Extra Trees	Word length, number of syllables, n-grams, frequency distribution	[6]
TMU	Random Forest	Word length, word frequency, probability features derived from corpora	[78]
UnibucKernel	Kernel-based learning with SVMs	Character n-grams, semantic features, and word embeddings	[34]

Table A.3. Systems Submitted to LCP–2021 in Alphabetical Order as Shown in [147]

Team	Classifiers	Features	Paper
Alejandro Mosquera	Gradient Boosted Regression	Length, Frequency, Semantic, Sentence	[108]
Andi	Ridge Regression, Gradient Boosted Regression	Psycholinguistic, Glove, Word2Vec, ConceptNet NumberBatch, BERT, RoBERTa, ELECTRA, ALBERT, DeBERTa	[137]
Archer	Random Forest Regression, Gradient Boosted Regression	Length, Frequency, Psycholinguistic, Scrabble Score, Word Inclusion, Semantic	[139]
BigGreen	Gradient Boosted Regression, BERT	Length, Semantic, Glove, Elmo, InferSent, Phonetic, Frequency, POS	[76]
C3SL	Multi-layer Perceptron	Sent2Vec	[9]
Cambridge	BERT, Random Forest Regression	Frequency, Syntactic, Length	[188]
CLULEX	Decision Tree	Frequency, POS, Named Entities, Word Inclusion, Sentence, Bert	[154]
CompNA	Decision Tree Ensemble	Length, Semantic, Glove, Word Inclusion	[168]
CS-UM6P	BERT, RoBERTa	Token and Context Encoded	[101]
CSECU-DSG	BERT, RoBERTa	Token and Context Encoded	[15]
DeepBlueAI	BERT, ALBERT, RoBERTa, ERNIE	Token and Context Encoded	[123]
IA PUCP	Gradient Boosted Regression	Sentence, POS, n-gram Frequency, RoBERTa, XLNet, BERT	[135]
IITK@LCP	Linear Regression, Support Vector Machine	ELECTRA + Glove	[150]
JCT	Gradient Boosted Regression	POS, Frequency, BERT, Cluster Features	[91]
JUST-BLUE	Average of Weighted Bert and Roberta	Token Encoded and Context Encoded	[183]
Katildakat	Linear Regression, Multi-layer Perceptron	BERT, Length, BERT-score, Frequency, Semantic	[169]
LAST	Gradient Boosted Regression	Frequency, Psycholinguistic, Sentence, Bigram Association	[22]
LCP-RIT	Random Forest Regressor	Length, Frequency, Character n-Grams, Psycholinguistic, POS	[51]
LRL_NC	Random Forest Regressor	Frequency, Semantic, Language Model, Psycholinguistic, Word Inclusion	[3]
Hub	RoBERTa, Inception	TF-IDF, Context Encoded	[75]
Manchester Metropolitan	CNN	Frequency, Psycholinguistic, Length, Embeddings	[57]
OCHADAI-KYOTO	BERT, RoBERTa	Token and Context Encoded	[165]
PolyU CBS-Comp	Gradient Boosted Regression	Frequency, Length, Capitalization, POS, Embeddings, BERT, GPT-2	[181]
RG PA	RoBERTa	Context Encoded	[130]
RS_GV	Feed-Forward Neural Network	GLoVE, ELMo, BERT, Flair, Readability, Length, Frequency, Semantic, Psycholinguistic, Morphological, Word Inclusion, Named Entity	[159]
Stanford MLab	Gradient Boosted Regression	Glove, Length, POS, Named Entity	[138]
TUDA-CCL	Gradient Boosted Regression	Linguistic, Semantic, Embeddings, Psycholinguistic, Frequencies, Word Inclusion	[62]
UNBNLP	Neural Network, Support Vector Machine	Length, Frequency, Character-Level-Encoder, BERT	[81]
UPB	BERT, RoBERTa, Linear Regression	Transformers, Word Embeddings, Character Embeddings, Length, Psycholinguistic	[190]
UTFPR	Support Vector Machine	Frequency, Length, Semantic, Bert Embedding	[118]

Table A.4. Datasets Used for English Complexity Prediction Research Arranged in Chronological Order

Dataset	Complexity	Size	Annotators	Noteworthy Comments	Paper
LS–2012	Comparative	201 complex words each with several candidate simplifications.	Native English speakers provided simplifications and four L2 learners ranked these simplifications based on their complexity.	Each Complex word is shown in 10 different contexts.	[155]
CW Corpus	Binary—Comparative	731 complex words and their equivalent simplification.	Complex words gained via Wikipedia edit history, editor comments, and a series of simplification checks.	Complex words are provided with context.	[143]
Horn et al. [74]	Comparative	500 complex words each with 50 candidate simplifications.	50 annotators from the US.	Data was acquired from the sentence-aligned Wikipedia corpus. Complex words are also provided with context.	[74]
CWI–2016	Binary	35,958 tokens with 232,481 instances; 3,854 of these tokens were labeled as complex.	400 non-native English speakers from a mix of international and educational backgrounds with varying levels of English proficiency.	Training set included 2,237 target words in 200 sentences, whereas the test set included 88,221 target words in 9,000 sentences.	[115]
CWI–2018	Binary—Continuous	34,789 English, 7,905 German, 17,605 Spanish, and 2,251 French words. Out of these, 14,428, 3,272, 7,015, and 657 were complex, respectively.	A mix of native and non-native speaking annotators for a variety of international backgrounds.	Datasets were also divided on source: News, WikiNews, and Wikipedia.	[185]
Word Complexity Lexicon	Continuous	15,000 words labeled with varying degrees of complexity.	11 non-native yet fluent English speakers.	Used a dictomous 6-point likert scale for annotation: very easy (very hard), moderately easy (moderately hard), and easy (hard).	[97]
Personalized LS Dataset	Personalized—Continuous	12,000 words labeled with varying degrees of complexity.	15 learners of English, who were native Japanese speakers.	Used a 5-point likert scale for annotation. Scores 1–4 were deemed as complex, whereas 5 was considered non-complex.	[89]
CompLex Dataset	Continuous	10,800 words and MWEs labeled with varying degrees of complexity.	Annotators were crowd sourced from the US, UK, and Australia. A median of 7 annotators labeled each word.	Used a 5-point likert scale for annotation. Scores were directly converted into continuous complex values. Words in context were taken from the Bible, biomedical articles, and europarl.	[146]

ACKNOWLEDGMENTS

The authors would like to thank Richard Evans for the valuable suggestions and feedback provided. We further thank the anonymous ACM CSUR reviewers for their insightful feedback.

REFERENCES

[1] Emad Abdallah, Alaa Eddien Abdallah, Ahmed F. Otoom, and Mohammad Bsoul. 2013. Simplified features for email authorship identification. *International Journal of Security and Networks* 8, 2 (2013), 71–81.

[2] Ahmed AbuRa'ed and Horacio Saggion. 2018. LaSTUS/TALN at complex word identification (CWI) 2018 shared task. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

[3] Raksha Agarwal and Niladri Chatterjee. 2021. LangResearchLab NC at SemEval-2021 Task 1: Linguistic feature based modelling for lexical complexity. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[4] Suha S. Al-Thanyyan and Aqil M. Azmi. 2021. Automated text simplification: A survey. *Computing Surveys* 54, 2 (2021), 1–36.

[5] Rodrigo Alarcón, Lourdes Moreno, and Paloma Martínez. 2020. Hulat - ALexS CWI task - CWI for language and learning disabilities applied to university educational texts. In *Proceedings of the 1st Edition of the ALexS Workshop*. Iberian Languages Evaluation Forum Proceedings.

[6] David Alfter and Ildikó Pilán. 2018. SB@GU at the complex word identification 2018 shared task. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

[7] Khaled M. Alhawiti. 2015. Innovative processes in computer assisted language learning. *International Journal of Advanced Research in Artificial Intelligence* 4, 2 (2015), 7–13.

[8] Ali Alkhatlan and Jugal Kalita. 2019. Intelligent tutoring systems: A comprehensive historical survey with recent developments. *International Journal of Computer Applications* 181, 43 (2019), 1–20.

[9] Raul Almeida, Hegler Tissot, and Marcos Fabro. 2021. C3SL at SemEval-2021 task 1: Predicting lexical complexity of words in specific contexts with sentence embeddings. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[10] Sandra Maria Aluísio and Caroline Gasperin. 2010. Fostering digital inclusion and accessibility: The PorSimples project for simplification of portuguese texts. In *Proceedings of YIWCALA Workshop at NAACL HLT'10*.

[11] Marilisa Amoia and Massimo Romanelli. 2012. SB: mmSystem—Using decompositional semantics for lexical simplification. In *SEM 2012: The 1st Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics.

[12] Jonathan Anderson. 1983. Lix and Rix: Variations on a little-known readability index. *Journal of Reading* 26, 6 (1983), 490–496.

[13] Alessio Palmero Aprosio, Stefano Menini, and Sara Tonelli. 2020. Adaptive complex word identification through false friend detection. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization (UMAP'20)*. ACM.

[14] Segun Taofeek Aroyehun, Jason Angel, Daniel Alejandro Pérez Alvarez, and Alexander Gelbukh. 2018. Complex word identification: Convolutional neural network vs. feature engineering. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

[15] Abdul Aziz, MD. Akram Hossain, and Abu Nowshed Chy. 2021. CSECU-DSG at SemEval-2021 task 1: Fusion of transformer models for lexical complexity prediction. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[16] Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William A. Baumgartner, K. Bretonnel Cohen, Karin Verspoor, Judith A. Blake, et al. 2012. Concept annotation in the CRAFT corpus. *BMC Bioinformatics* 13, 1 (2012), 161.

[17] Susana Bautista, Raquel Hervás, Pablo Gervás, Richard Power, and Sandra Williams. 2011. Experimental identification of the use of hedges in the simplification of numerical expressions. In *Proceedings of the 2nd Workshop on Speech and Language Processing for Assistive Technologies*. Association for Computational Linguistics.

[18] Susana Bautista, Raquel Hervás, Pablo Gervás, and Javier Rojo. 2017. An approach to treat numerical information in the text simplification process. *Universal Access in the Information Society* 16, 1 (2017), 85–102.

[19] Susana Bautista, Raquel Hervás, Pablo Gervás, Richard Power, and Sandra Williams. 2011. How to make numerical information accessible: Experimental identification of simplification strategies. In *IFIP Conference on Human-Computer Interaction* 1 (2011), 57–64.

[20] Susana Bautista and Horacio Saggion. 2014. Can numerical expressions be simpler? Implementation and demostration of a numerical simplification system for Spanish. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA).

[21] Michael R. Berthold, Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Kilian Thiel, and Bernd Wiswedel. 2009. KNIME—the Konstanz information miner: Version 2.0 and beyond. *ACM SIGKDD Explorations Newsletter* 11, 1 (2009), 26–31.

[22] Yves Bestgen. 2021. LAST at SemEval-2021 task 1: Improving multi-word complexity prediction using bigram association measures. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[23] Mokhtar B. Billami, Thomas François, and Núria Gala. 2018. ReSyf: A French lexicon with ranked synonyms. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics.

[24] Joachim Bingel and Johannes Bjerva. 2018. Cross-lingual complex word identification with multitask learning. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

[25] Joachim Bingel, Natalie Schluter, and Héctor Martínez Alonso. 2016. CoastalCPH at SemEval'16 task 11: The importance of designing your neural networks right. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[26] Tudor Boran, Muhamet Martinaj, and Md Shafaeat Hossain. 2020. Authorship identification on limited samplings. *Computers & Security* 97 (2020), 101943.

[27] Stefan Bott, Luz Rello, Biljana Drndarevic, and Horacio Saggion. 2012. Can Spanish be simpler? LexSiS: Lexical simplification for Spanish. In *Proceedings of COLING 2012*. The COLING 2012 Organizing Committee, 357–374.

[28] Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram version 1. *Linguistic Data Consortium (LDC'06)*.

[29] Chris Brew and D. McKelvie. 1996. Word-pair extraction for lexicography. *Word-Pair Extraction for Lexicography*. 45–55.

[30] Julian Brooke, Alexandra Uitdenbogerd, and Timothy Baldwin. 2016. Melbourne at SemEval 2016 task 11: Classifying type-level word complexity using random forests with corpus and word list features. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[31] Marc Brysbaert and Boris New. 2009. Moving beyond Kučera and Francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English. *Behavior Research Methods* 41, 4 (2009), 977–990.

[32] Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2013. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods* 46 (2013), 904–911.

[33] Bram Bulté and Alex Housen. 2012. Defining and operationalising L2 complexity. *Dimensions of L2 Performance and Proficiency*. John Benjamins, 21–46.

[34] Andrei Butnaru and Radu Tudor Ionescu. 2018. UnibucKernel: A kernel-based learning method for complex word identification. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

[35] John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. Practical simplification of English newspaper text to assist aphasic readers. In *Proceedings of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*.

[36] Jeanne Sternlicht Chall and Edgar Dale. 1995. *Readability Revisited: The New Dale-Chall Readability Formula*. Brookline Books.

[37] Xiaoman Chen and Xiaomin Xiao. 2019. An autonomous training about college students' reading of all-capital-letter English. *American Journal of Education and Information Technology* 3 (2019), 47–51.

[38] Prafulla Choubey and Shubham Pateria. 2016. Garuda & Bhasha at SemEval-2016 task 11: Complex word identification using aggregated learning models. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[39] Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: The Bible in 100 languages. *Language Resources and Evaluation* 49, 2 (2015), 375–395.

[40] Kevin Clark, Minh-Thang Luong, Quoc Le, and Christopher Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of the 8th International Conference on Learning Representations*. ICLR.

[41] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online.

[42] British National Corpus. 2015. http://www.natcorp.ox.ac.uk.

[43] The Wikipedia Corpus. 2001. https://www.english-corpora.org/wiki/.

[44] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning* 20, 1 (1995), 273–297.

[45] William Coster and David Kauchak. 2011. Simple English Wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

[46] Scott A. Crossley, Max M. Louwerse, Philip M. McCarthy, and Danielle S. McNamara. 2007. A linguistic analysis of simplified and authentic texts. *The Modern Language Journal* 91, 1 (2007), 15–30.

[47] Östen Dahl. 2004. *The Growth and Maintenance of Linguistic Complexity*. John Benjamins.

[48] Elnaz Davoodi and Leila Kosseim. 2016. CLaC at SemEval-2016 task 11: Exploring linguistic and psycho-linguistic features for complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[49] Jan De Belder and Marie-Francine Moens. 2010. Text simplification for children. In *Proceedings of the SIGIR Workshop on Accessible Search Systems*. ACM.

[50] Dirk De Hertog and Anaïs Tack. 2018. Deep learning architecture for complex word identification. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

[51] Abhinandan Desai, Kai North, Marcos Zampieri, and Christopher Homan. 2021. LCP-RIT at SemEval-2021 task 1: Exploring linguistic features for lexical complexity prediction. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[52] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. 4171–4186.

[53] Siobhan Devlin and John Tait. 1998. The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic Databases* (1998), 161–173.

[54] Ehsan Fathi and Babak Maleki Shoja. 2018. Deep neural networks for natural language processing. *Computational Analysis and Understanding of Natural Languages: Principles, Methods and Applications*. Handbook of Statistics, Vol. 38. Elsevier, 229–316.

[55] Pierre Finnimore, Elisabeth Fritzsch, Daniel King, Alison Sneyd, Aneeq Ur Rehman, Fernando Alva-Manchego, and Andreas Vlachos. 2019. Strong baselines for complex word identification across multiple languages. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

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

[57] Robert Flynn and Matthew Shardlow. 2021. Manchester metropolitan at SemEval-2021 task 1: Convolutional networks for complex word identification. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[58] Aina Garí Soler, Marianna Apidianaki, and Alexandre Allauzen. 2018. A comparative study of word embeddings and other features for lexical complexity detection in French. In *Actes de la Conférence TALN*.

[59] Caroline Gasperin, Lucia Specia, Tiago F. Pereira, and Sandra M. Aluisio. 2009. Learning when to simplify sentences for natural text simplification. *Encontro Nacional de Inteligencia Artificial* (2009), 809–818.

[60] Nat Gillin. 2016. Sensible at SemEval-2016 task 11: Neural nonsense mangled in ensemble mess. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[61] Anthony Gillioz, Jacky Casas, Elena Mugellini, and Omar Abou Khaled. 2020. Overview of the transformer-based models for NLP tasks. In *Proceedings of the Federated Conference on Computer Science and Information Systems*. FedCSIS.

[62] Sebastian Gombert and Sabine Bartsch. 2021. TUDA-CCL at SemEval-2021 task 1: Using gradient-boosted regression tree ensembles trained on a heterogeneous feature set for predicting lexical complexity. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[63] Sian Gooding. 2022. On the ethical considerations of text simplification. In *9th Workshop on Speech and Language Processing for Assistive Technologies (SLPAT'22)*. Association for Computational Linguistics.

[64] Sian Gooding and Ekaterina Kochmar. 2018. CAMB at CWI shared task 2018: Complex word identification with ensemble-based voting. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

[65] Sian Gooding and Ekaterina Kochmar. 2019. Complex word identification as a sequence labelling task. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

[66] Sian Gooding, Ekaterina Kochmar, Advait Sarkar, and Alan Blackwell. 2019. Comparative judgments are more consistent than binary classification for labelling word complexity. In *Proceedings of the 13th Linguistic Annotation Workshop*. Association for Computational Linguistics.

[67] Sian Gooding, Shiva Taslimipoor, and Ekaterina Kochmar. 2020. Incorporating multiword expression in phrase complexity estimation. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI'20)*. European Language Resources Association.

[68] Arthur Graesser, Xiangen Hu, Benjamin Nye, Kurt VanLehn, Rohit Kumar, Cristina Heffernan, Neil Heffernan, Beverly Woolf, Andrew Olney, Vasile Rus, Frank Andrasik, Philip Pavlik, Zhiqiang Cai, Jon Wetzel, Brent Morgan, Andrew Hampton, Anne Lippert, Lijia Wang, Qinyu Cheng, Joseph Vinson, Craig Kelly, Cadarrius McGlown, Charvi

Majmudar, Bashir Morshed, and Whitney Baer. 2018. ElectronixTutor: An intelligent tutoring system with multiple learning resources for electronics. *International Journal of STEM Education* 5, 15 (2018), 1–21.

[69] Robert Gunning. 1952. *The Technique of Clear Writing*. McGraw-Hill, New York.

[70] Gavin Hackeling. 2014. *Mastering Machine Learning with Scikit-Learn*. Packt Publishing.

[71] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *SIGKDD Explorations Newsletter* 11 (2009), 10–18.

[72] Hanban. 2014. *International Curriculum for Chinese Language Education*. Beijing Language and Culture University Press, Beijing, China.

[73] Nathan Hartmann and Leandro Borges dos Santos. 2018. NILC at CWI 2018: Exploring feature engineering and feature learning. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

[74] Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a lexical simplifier using Wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics.

[75] Bo Huang, Yang Bai, and Xiaobing Zhou. 2021. hub at SemEval-2021 task 1: Fusion of sentence and word frequency to predict lexical complexity. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[76] Aadil Islam, Weicheng Ma, and Soroush Vosoughi. 2021. BigGreen at SemEval-2021 task 1: Lexical complexity prediction with assembly models. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[77] Sujay Kumar Jauhar and Lucia Specia. 2012. UOW-SHEF: SimpLex—lexical simplicity ranking based on contextual and psycholinguistic features. In **SEM 2012: The 1st Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval'12)*. Association for Computational Linguistics.

[78] Tomoyuki Kajiwara and Mamoru Komachi. 2018. Complex word identification based on frequency in a learner corpus. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

[79] Tomoyuki Kajiwara, Hiroshi Matsumoto, and Kazuhide Yamamoto. 2013. Selecting proper lexical paraphrase for children. In *Proceedings of the 25th Conference on Computational Linguistics and Speech Processing (ROCLING'13)*. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

[80] David Kauchak. 2016. Pomona at SemEval-2016 task 11: Predicting word complexity based on corpus frequency. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[81] Milton King, Ali Hakimi Parizi, Samin Fakharian, and Paul Cook. 2021. UNBNLP at SemEval-2021 task 1: Predicting lexical complexity with masked language models and character-level encoders. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[82] Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*.

[83] Michal Konkol. 2016. UWB at SemEval-2016 task 11: Exploring features for complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[84] Reno Kriz, Eleni Miltsakaki, Marianna Apidianaki, and Chris Callison-Burch. 2018. Simplification using paraphrases and context-based lexical substitution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics.

[85] Miroslav Kubát and Jiří Milička. 2013. Vocabulary richness measure in genres. *Journal of Quantitative Linguistics* 20, 4 (2013), 339–349.

[86] Onur Kuru. 2016. AI-KU at SemEval-2016 task 11: Word embeddings and substring features for complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[87] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of the 8th International Conference on Learning Representations*. Addis Ababa, Ethiopia, Online.

[88] John Lee and Chak Yan Yeung. 2018. Automatic prediction of vocabulary knowledge for learners of Chinese as a foreign language. In *Proceedings of the 2nd International Conference on Natural Language and Speech Processing (ICNLSP'18)*.

[89] John Lee and Chak Yan Yeung. 2018. Personalizing lexical simplification. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics.

[90] Geoffrey Leech, Paul Rayson, et al. 2014. *Word Frequencies in Written and Spoken English: Based on the British National Corpus*. Routledge.

[91] Chaya Liebeskind, Otniel Elkayam, and Shmuel Liebeskind. 2021. JCT at SemEval-2021 task 1: Context-aware representation for lexical complexity prediction. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[92] Anne-Laure Ligozat, Cyril Grouin, Anne Garcia-Fernandez, and Delphine Bernhard. 2012. ANNLOR: A naïve notation-system for lexical outputs ranking. In *SEM 2012: The 1st Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics.

[93] Linyan Liu, John W. Schwieter, Fenqi Wang, and Huanhuan Liu. 2022. First and second languages differentially affect rationality when making decisions: An ERP study. *Biological Psychology* 169 (2022), 108265.

[94] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR, abs/1907.11692* (2019).

[95] Kenneth Logan and M. J. Kieffer. 2017. Evaluating the role of polysemous word knowledge in reading comprehension among bilingual adolescents. *Reading and Writing* 30 (2017), 1687–1704.

[96] Bernard Lété, Liliane Sprenger-Charolles, and Pascale Colé. 2004. Manulex: A grade-level lexical database from French elementary-school readers. *Behavior Research Methods, Instruments, & Computers* 36 (2004), 156–166.

[97] Mounica Maddela and Wei Xu. 2018. A word-complexity lexicon and a neural readability ranking model for lexical simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

[98] Kikuo Maekawa, Makoto Yamazaki, Takehiko Maruyama, Masaya Yamaguchi, Hideki Ogura, Wakako Kashino, Toshinobu Ogiso, Hanae Koiso, and Yasuharu Den. 2010. Design, compilation, and preliminary analyses of balanced corpus of contemporary written Japanese. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA).

[99] Shervin Malmasi, Mark Dras, and Marcos Zampieri. 2016. LTG at SemEval-2016 task 11: Complex word identification with classifier ensembles. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[100] Shervin Malmasi and Marcos Zampieri. 2016. MAZA at SemEval-2016 task 11: Detecting lexical complexity using a decision stump meta-classifier. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[101] Nabil El Mamoun, Abdelkader El Mahdaouy, Abdellah El Mekki, Kabil Essefar, and Ismail Berrada. 2021. CS-UM6P at SemEval-2021 task 1: A deep learning model-based pre-trained transformer encoder for lexical complexity. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[102] José Manuel Martínez Martínez and Liling Tan. 2016. USAAR at SemEval-2016 task 11: Complex word identification with sense entropy and sentence perplexity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[103] Harry McLaughlin. 1969. SMOG grading—a new readability formula. *Journal of Reading* 22 (1969), 639–646.

[104] Borbor Merejildo. 2021. *Creación de un Corpus de Textos Universitarios en Español para la Identificación de Palabras Complejas en el Área de la Simplificación Léxica*. Master's Thesis. Universidad de Guayaquil.

[105] Merriam-Webster. 2020. Engaged.

[106] George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM* 38, 11 (1995), 39–41.

[107] Neil P. Morris, Mariya Ivancheva, Taryn Coop, Rada Mogliacci, and Bronwen Swinnerton. 2020. Negotiating growth of online education in higher education. *International Journal of Educational Technology in Higher Education* 17, 48 (2020), 1–16.

[108] Alejandro Mosquera. 2021. Alejandro Mosquera at SemEval-2021 task 1: Exploring sentence and word features for lexical complexity prediction. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[109] Ahmed Mousa. 2015. Acquisition of the closing diphthongs /@u/ and /eI/ in English L2 and Jamaican Creole: A comparative study. *SAGE Open* 5, 2 (2015), 1–15.

[110] Niloy Mukherjee, Braja Gopal Patra, Dipankar Das, and Sivaji Bandyopadhyay. 2016. JU_NLP at SemEval-2016 task 11: Identifying complex words in a sentence. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[111] Daiki Nishihara and Tomoyuki Kajiwara. 2020. Word complexity estimation for Japanese lexical simplification. In *Proceedings of the 12th Language Resources and Evaluation Conference*. European Language Resources Association.

[112] Kai North, Marcos Zampieri, and Matt Shardlow. 2018. An evaluation of binary comparative lexical complexity models. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA'22)*.

[113] University of Tsukuba. 2014. *NLT Corpus*. National Institute for Japanese Language and Linguistics, Lago Institute of Language: NINJAL-LWP for TWC.

[114] Lourdes Ortega. 2003. Syntactic complexity measures and their relationship to L2 proficiency: A research synthesis of college-level L2 writing. *Applied Linguistics* 24, 4 (12 2003), 492–518.

[115] Gustavo Paetzold and Lucia Specia. 2016. SemEval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[116] Gustavo Paetzold and Lucia Specia. 2016. SV000gg at SemEval-2016 task 11: Heavy gauge complex word identification with system voting. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[117] Gustavo Paetzold and Lucia Specia. 2017. Lexical simplification with neural ranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

[118] Gustavo Henrique Paetzold. 2021. UTFPR at SemEval-2021 task 1: Complexity prediction by combining BERT vectors and classic features. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[119] Gustavo H. Paetzold and Lucia Specia. 2013. Text simplification as tree transduction. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*.

[120] Gustavo H. Paetzold and Lucia Specia. 2017. A survey on lexical simplification. *Journal of Artificial Intelligence Research* 60, 1 (Sept. 2017), 549–593.

[121] Ashish Palakurthi and Radhika Mamidi. 2016. IIIT at SemEval-2016 task 11: Complex word identification using nearest centroid classification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[122] Gabriele Pallotti. 2015. A simple view of linguistic complexity. *Second Language Research* 31, 1 (2015), 117–134.

[123] Chunguang Pan, Bingyan Song, Shengguang Wang, and Zhipeng Luo. 2021. DeepBlueAI at SemEval-2021 task 1: Lexical complexity prediction with a deep ensemble approach. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[124] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*.

[125] Sarah Petersen and Mari Ostendorf. 2009. A machine learning approach to reading level assessment. *Computer Speech & Language* 23, 1 (2009), 89–106.

[126] Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

[127] Maja Popović. 2018. Complex word identification using character n-grams. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

[128] Diana Pulido. 2004. The relationship between text comprehension and second language incidental vocabulary acquisition: A matter of topic familiarity? *Language Learning* 54, 3 (2004), 469–523.

[129] Maury Quijada and Julie Medero. 2016. HMC at SemEval-2016 task 11: Identifying complex words using depth-limited decision trees. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[130] Gang Rao, Maochang Li, Xiaolong Hou, Lianxin Jiang, Yang Mo, and Jianping Shen. 2021. RG PA at SemEval-2021 task 1: A contextual attention-based model with RoBERTa for lexical complexity prediction. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[131] Luz Rello, Ricardo Baeza-Yates, Laura Dempere-Marco, and Horacio Saggion. 2013. Frequent words improve readability and short words improve understandability for people with dyslexia. In *Proceedings of the INTERACT 2013: 14th IFIP TC13 Conference on Human-Computer Interaction*.

[132] Luz Rello, Susana Bautista, Ricardo Baeza-Yates, Pablo Gervás, Raquel Hervás, and Horacio Saggion. 2013. One half or 50%? An eye-tracking study of number representation readability. In *Human-Computer Interaction – INTERACT 2013*, Paula Kotzé, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler (Eds.). Springer, Berlin.

[133] Irina Rets and Jekaterina Rogaten. 2020. To simplify or not? Facilitating English L2 users' comprehension and processing of open educational resources in English using text simplification. *Journal of Computer Assisted Learning* 37, 3 (2020), 705–717.

[134] Andi Rexha, Mark Kröll, Hermann Ziak, and Roman Kern. 2018. Authorship identification of documents with high content similarity. *Scientometrics* 115 (2018), 223–237.

[135] Kervy Rivas Rojas and Fernando Alva-Manchego. 2021. IAPUCP at SemEval-2021 task 1: Stacking fine-tuned transformers is almost all you need for lexical complexity prediction. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[136] Francesco Ronzano, Ahmed Abura'ed, Luis Espinosa Anke, and Horacio Saggion. 2016. TALN at SemEval-2016 task 11: Modelling complex words by contextual, lexical and semantic features. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[137] Armand Rotaru. 2021. ANDI at SemEval-2021 task 1: Predicting complexity in context using distributional models, behavioural norms, and lexical resources. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[138] Erik Rozi, Niveditha Iyer, Gordon Chi, Enok Choe, Kathy Lee, Kevin Liu, Patrick Liu, Zander Lack, Jillian Tang, and Ethan A. Chi. 2021. Stanford MLab at SemEval-2021 task 1: Tree-based modelling of lexical complexity using word embeddings. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[139] Irene Russo. 2021. archer at SemEval-2021 task 1: Contextualising lexical complexity. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[140] S. P. Sanjay, Kumar M. Anand, and K. P. Soman. 2016. AmritaCEN at SemEval-2016 task 11: Complex word identification using word embedding. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. Association for Computational Linguistics.

[141] Kuntum Palupi Setyaningsih, Agus Wijayanto, and Suparno Suparno. 2019. English vowels and diphthongs problems of Sundanese learners. *ELS Journal on Interdisciplinary Studies in Humanities* 2, 4 (2019), 571–581.

[142] Matthew Shardlow. 2013. A comparison of techniques to automatically identify complex words. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*. Association for Computational Linguistics.

[143] Matthew Shardlow. 2013. The CW corpus: A new resource for evaluating the identification of complex words. In *Proceedings of the 2nd Workshop on Predicting and Improving Text Readability for Target Reader Populations*. Association for Computational Linguistics.

[144] Matthew Shardlow. 2014. Out in the open: Finding and categorising errors in the lexical simplification pipeline. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA).

[145] Matthew Shardlow. 2014. Out in the open: Finding and categorising errors in the lexical simplification pipeline. In *Proceedings of Language Resources and Evaluation Conference (LREC'14)*. 1583–1590.

[146] Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. CompLex—A new corpus for lexical complexity prediction from Likert scale data. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI'20)*. European Language Resources Association.

[147] Matthew Shardlow, Richard Evans, Gustavo Paetzold, and Marcos Zampieri. 2022. SemEval-2021 task 1: Lexical complexity prediction. In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval'21)*.

[148] Matthew Shardlow, Richard Evans, and Marcos Zampieri. 2021. Predicting lexical complexity in English texts. *Language Resources and Evaluation* 56, 1 (2021).

[149] Kim Cheng Sheang. 2019. Multilingual complex word identification: Convolutional neural networks with morphological and linguistic features. In *Proceedings of the Student Research Workshop Associated with RANLP 2019*. INCOMA Ltd.

[150] Neil Shirude, Sagnik Mukherjee, Tushar Shandhilya, Ananta Mukherjee, and Ashutosh Modi. 2021. IITK@LCP at SemEval-2021 task 1: Classification for lexical complexity regression task. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[151] Punardeep Sikka and Vijay Mago. 2020. A survey on text simplification. *CoRR/abs/2008.08612* 165, 2 (2020), 259–298.

[152] Ravi Sinha. 2012. UNT-SimpRank: Systems for lexical simplification ranking. In **SEM 2012: The 1st Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval'12)*. Association for Computational Linguistics, 493–496.

[153] Greta Smolenska. 2018. *Complex Word Identification for Swedish*. Master's Thesis. Uppsala University, Sweden.

[154] Greta Smolenska, Peter Kolb, Sinan Tang, Mironas Bitinis, Héctor Hernández, and Elin Asklöv. 2021. CLULEX at SemEval-2021 task 1: A simple system goes a long way. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[155] Lucia Specia, Kumar Jauhar, Sujay, and Rada Mihalcea. 2012. SemEval - 2012 task 1: English lexical simplification. In *Proceedings of the 1st Joint Conference on Lexical and Computational Semantics (*SEM)*. Association for Computational Linguistics.

[156] L. Srinivasan and Chinnu Nalini. 2019. An improved framework for authorship identification in online messages. *Cluster Computing* 22, 1 (2019), 12101–12110.

[157] Tim Stagich. 1995. Cultural context: The key to second language learning and acquisition. *Educational Horizons* 73, 2 (1995), 59–61.

[158] Sanja Štajner and Maja Popović. 2019. Automated text simplification as a preprocessing step for machine translation into an under-resourced language. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'19)*. INCOMA Ltd.

[159] Regina Stodden and Gayatri Venugopal. 2021. RS_GV at SemEval-2021 task 1: Sense relative lexical complexity prediction. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[160] Antonio Rico Sulayes. 2020. General lexicon-based complex word identification extended with stem N-grams and morphological engines. In *Proceedings of the 1st Edition of the ALexS Workshop*. Iberian Languages Evaluation Forum Proceedings.

[161] Anaïs Tack. 2021. *Mark My Words! On the Automated Prediction of Lexical Difficulty for Foreign Language Readers*. Ph.D. Dissertation. Université Catholique de Louvain, France.

[162] Anaïs Tack, Thomas François, Anne-Laure Ligozat, and Cédrick Fairon. 2016. Evaluating lexical simplification and vocabulary knowledge for learners of French: Possibilities of using the FLELex resource. In *Proceedings of the T10th International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA).

[163] Anaïs Tack, Thomas François, Anne-Laure Ligozat, and Cédrick Fairon. 2016. Modèles adaptatifs pour prédire automatiquement la compétence lexicale d'un apprenant de français langue étrangère (Adaptive models for automatically predicting the lexical competence of French as a foreign language learners). In *Actes de la Conférence Conjointe JEP-TALN-RECITAL 2016*. AFCP - ATALA.

[164] Ludovic Tanguy, Franck Sajous, Basilio Calderone, and Nabil Hathout. Authorship attribution: Using rich linguistic features when training data is scarce. In *Proceedings of the 6th International Conference of CLEF on Experimental IR meets Multilinguality, Multimodality, and Interaction (CLEF'15)*.

[165] Yuki Taya, Lis Kanashiro Pereira, Fei Cheng, and Ichiro Kobayashi. 2021. OCHADAI-KYODAI at SemEval-2021 task 1: Enhancing model generalization and robustness for lexical complexity prediction. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[166] Sheng-Shiang Tseng and Hui-Chin Yeh. 2019. Fostering EFL teachers' CALL competencies through project-based learning. *Educational Technology & Society* 22, 1 (2019), 94–105.

[167] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017), 5998–6008.

[168] Giuseppe Vettigli and Antonio Sorgente. 2021. CompNA at SemEval-2021 task 1: Prediction of lexical complexity analyzing heterogeneous features. In *Proceedings of the 15th Workshop on Semantic Evaluation*. Association for Computational Linguistics.

[169] Katja Voskoboinik. 2021. katildakat at SemEval-2021 task 1: Lexical complexity prediction of single words and multiword expressions in English. In *Proceedings of the 15th Workshop Linguistics on Semantic Evaluation*. Association for Computational.

[170] Sanja Štajner and Maja Popović. 2016. Can text simplification help machine translation? *Baltic Journal of Modern Computing* 4, 2 (2016), 230–242.

[171] Robert A. Wagner and Michael J. Fischer. 1974. The string-to-string correction problem. *Journal of the ACM* 21, 1 (1974), 168–173.

[172] Maolin Wang, Shervin Malmasi, and Mingxuan Huang. 2015. The Jinan Chinese learner corpus. In *Proceedings of the 10th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

[173] Nikhil Wani, Sandeep Mathias, Jayashree Aanand Gajjam, and Pushpak Bhattacharyya. 2018. The whole is greater than the sum of its parts: Towards the effectiveness of voting ensemble classifiers for complex word identification. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

[174] William Massami Watanabe, Arnaldo Candido, M. A. Amâncio, M. De Oliveira, T. A. S. Pardo, R. P. M. Fortes, and S. M. Aluísio. 2010. Adapting Web content for low-literacy readers by using lexical elaboration and named entities labeling. *New Review of Hypermedia and Multimedia* 16, 3 (2010), 303–327.

[175] Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria Aluísio. 2009. Facilita: Reading assistance for low-literacy readers. Association for Computing Machinery, New York.

[176] Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. Association for Computational Linguistics.

[177] Michael Wilson. 1988. MRC psycholinguistic database: Machine-usable dictionary, version 2.00. *Behavior Research Methods, Instruments, & Computers* 20, 1 (1988), 6–10.

[178] William E. Winkler. 1990. String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage. In *Proceedings of the Section on Survey Research*.

[179] Krzysztof Wróbel. 2016. PLUJAGH at SemEval-2016 task 11: Simple system for complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16).* Association for Computational Linguistics.

[180] Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, and Bo Long. 2021. Graph neural networks for natural language processing: A survey. arXiv.

[181] Rong Xiang, Jinghang Gu, Emmanuele Chersoni, Wenjie Li, Qin Lu, and Chu-Ren Huang. 2021. PolyU CBS-Comp at SemEval-2021 task 1: Lexical complexity prediction (LCP). In *Proceedings of the 15th Workshop on Semantic Evaluation.* Association for Computational Linguistics.

[182] Man Yang, Cooc North, and Li Sheng. 2017. An investigation of cross-linguistic transfer between Chinese and English: A meta-analysis. *Asian-Pacific Journal of Second Foreign Language Education* 2, 15 (2017), 1–21.

[183] Tuqa Bani Yaseen, Qusai Ismail, Sarah Al-Omari, Eslam Al-Sobh, and Malak Abdullah. 2021. JUST-BLUE at SemEval-2021 task 1: Predicting lexical complexity using BERT and RoBERTa pre-trained language models. In *Proceedings of the 15th Workshop on Semantic Evaluation.* Association for Computational Linguistics.

[184] Chak Yan Yeung and John Lee. 2018. Personalized text retrieval for learners of Chinese as a foreign language. In *Proceedings of the 27th International Conference on Computational Linguistics.* Association for Computational Linguistics.

[185] Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications.* Association for Computational Linguistics.

[186] Seid Muhie Yimam, Sanja Štajner, Martin Riedl, and Chris Biemann. 2017. Multilingual and cross-lingual complex word identification. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP'17).* INCOMA Ltd.

[187] Rongmei Yu. 2020. Culture in second or foreign language acquisition. *Journal of Language Teaching and Research* 11, 6 (2020), 943–947.

[188] Zheng Yuan, Gladys Tyen, and David Strohmaier. 2021. Cambridge at SemEval-2021 task 1: An ensemble of feature-based and neural models for lexical complexity prediction. In *Proceedings of the 15th Workshop on Semantic Evaluation.* Association for Computational Linguistics.

[189] George-Eduard Zaharia, Dumitru-Clementin Cercel, and M. Dascalu. 2020. Cross-lingual transfer learning for complex word identification. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI'20).*

[190] George-Eduard Zaharia, Dumitru-Clementin Cercel, and Mihai Dascalu. 2021. UPB at SemEval-2021 task 1: Combining deep learning and hand-crafted features for lexical complexity prediction. In *Proceedings of the 15th Workshop on Semantic Evaluation.* Association for Computational Linguistics.

[191] Farooq Zaman, Matthew Shardlow, Saeed-Ul Hassan, Naif Radi Aljohani, and Raheel Nawaz. 2020. HTSS: A novel hybrid text summarisation and simplification architecture. *Information Processing and Management* 57, 102351 (2020), 1–13.

[192] Jenny Alexandra Ortiz Zambrano and Arturo Montejo Ráez. 2020. Overview of ALexS 2020: First workshop on lexical analysis at SEPLN. In *Proceedings of the 1st Edition of the ALexS Workshop.* Iberian Languages Evaluation Forum Proceedings.

[193] Marcos Zampieri, Shervin Malmasi, Gustavo Paetzold, and Lucia Specia. 2017. Complex word identification: Challenges in data annotation and system performance. In *Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA'17).* Asian Federation of Natural Language Processing.

[194] Marcos Zampieri, Liling Tan, and Josef van Genabith. 2016. MacSaar at SemEval-2016 task 11: Zipfian and character features for ComplexWord identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16).* Association for Computational Linguistics.

[195] Qing Zeng, Eunjung Kim, Jon Crowell, and Tony Tse. 2005. A text corpora-based estimation of the familiarity of health terminology. Springer-Verlag, Berlin.

[196] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics.

[197] Elena Zotova, Montse Cuadros, Naiara Pérez, and Aitor García-Pablos. 2020. Vicomtech at ALexS 2020: Unsupervised complex word identification based on domain frequency. In *Proceedings of the 1st Edition of the ALexS Workshop.* Iberian Languages Evaluation Forum Proceedings.

Received 30 June 2021; revised 21 July 2022; accepted 2 August 2022