Manchester Metropolitan University

e-space

Manchester Metropolitan University's Research Repository

Please cite the Published Version

North, Kai, Zampieri, Marcos and Shardlow, Matthew (2023) Lexical complexity prediction: an overview. ACM Computing Surveys, 55 (9). p. 179. ISSN 0360-0300

DOI: https://doi.org/10.1145/3557885

Publisher: Association for Computing Machinery

Version: Accepted Version

Downloaded from: https://e-space.mmu.ac.uk/631704/

Additional Information: © Association for Computing Machinery, 2023. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in ACM Computing Surveys, http://dx.doi.org/10.1145/3557885

Enquiries:
If you have questions about this document, contact openresearch@mmu.ac.uk. Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines)

Lexical Complexity Prediction: An Overview

KAI NORTH, Rochester Institute of Technology, USA
MARCOS ZAMPIERI, Rochester Institute of Technology, 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 modelling has been applied to identify complex words in texts and substitute them for simpler alternatives. In this paper, 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 also 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: • **General and reference** → **Surveys and overviews**.

Additional Key Words and Phrases: Complex Word Identification, Lexical Complexity Prediction, NLP, Lexical Simplification, Text Simplification, Assistive Technologies.

1 INTRODUCTION

Understanding the meaning of words in context is fundamental for reading comprehension. The perceived complexity, or difficulty, of a target word within a given text varies widely among readers. With an increased demand for distance learning and educational technologies [90], research into automatically predicting which words are likely to cause comprehension problems is becoming a popular area of research [95, 119, 151]. Systems have been created to identify complex words for second-language learners [84], children [67], people suffering from a reading disability, such as dyslexia [107] or aphasia [27, 43], or more generally, individuals with low literacy [48, 142].

In Computational Linguistics and Natural Language Processing (NLP), the task of automatically recognizing complex words is most often achieved by training machine learning (ML) classifiers. These ML classifiers assign a complexity value to each target word within a given text. This task is commonly referred to as Complex Word

Authors' addresses: Kai North, kn1473@rit.edu, Rochester Institute of Technology, USA; Marcos Zampieri, Rochester Institute of Technology, USA; Matthew Shardlow, Manchester Metropolitan University, UK.

Identification (CWI) [95]. However, in recent years, Lexical Complexity Prediction (LCP) has been proposed as the overarching term for all complexity prediction research[117, 119][1].

In Linguistics, the term *complex* is used to denote a word that comprises of two or more morphemes, such as *"un–believ–able"* or *"in–evit–able"*. Within LCP, the term *complex* is used as a "synonym for difficulty" [85]. A complexity value defines the level of difficulty a given target population (or individual) may have in understanding a target word. This complexity value is typically informed by those that have manually annotated the datasets used to train LCP systems.

There exists a fundamental difference between previous CWI and current LCP research [2]. The former entails a binary classification task, whereas the latter involves a regression based task that relies on multi-labeled data. CWI requires its annotators to assign two labels that correspond to two complexity values: complex (1) or non-complex (0). These values are then used to train a binary CWI system.

LCP is alternatively a regression based task that models complexity as a continuum. An LCP system relies on a dataset annotated using a likert scale rather than binary annotation. A likert scale provides annotators with multiple labels which are then later used to generate continuous complexity values between 0 and 1 (Section 2). Within LCP, a target word's final complexity value is determined as the average complexity values of the labels assigned by its annotators. This results in complexity assignments such as that shown in the following extract taken from the CompLex dataset [117] (See Table 1).

Extract:	**Folly**	is	**set**	in	**great dignity**
Complexity value:	0.57	is	0.18	in	0.15

Table 1. Example of a sentence annotated with complexity values. Target words are in bold.

The target word "*folly*" has a complexity value of 0.57 [117]. This makes it an example of a complex word, since it has a perceived level of difficulty between neutral (0.5) and difficult (0.75) [82, 117]. Being archaic and no longer used within everyday speech [113], many individuals may not be familiar with this target word, and thus unable to interpret its meaning. The purpose of an LCP system is to correctly assign target words like "*folly*", with a complexity value that accurately reflects their perceived level of difficulty, without the need for human judgement.

This survey provides the reader with a comprehensive overview of LCP literature, with a particular focus on the work carried out in the last 10 years. It describes LCP's origins within lexical simplification and CWI, to its latest developments (Section 2), shared-tasks (Section 3), and state-of-the-art systems (Section 4). 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 [90]. It also comes at a time of diversification, with LCP interacting with other topics in NLP, such as sentiment analysis [29, 69], authorship identification [1, 128], and machine translation [138]. 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, advancements, and current baselines needed to develop LCP further.

This survey has the following structure. Section 2 outlines the different types of lexical complexity prediction, ranging from comparative, binary, continuous, and personalized to predicting the complexity of multi-word expressions. Section 3 details the international competitions that challenged participating teams with the development of LCP systems: CWI–2016 [95], CWI–2018 [151], and LCP–2021 [118]. Section 4 provides a historical

[1]In this paper, we will be using LCP as the overarching term and CWI specifically when we refer to the binary task of complexity prediction (Section 2) [117, 119].

[2]The fundamental difference between CWI and LCP research is further discussed in Section 2.

overview of the models used for LCP, including feature engineering approaches, neural networks to state-of-the-art transformer-based models. Section 5 demonstrates LCP's use cases and applications. Section 6 gives an overview of the datasets and resources used for LCP as well as LCP in languages other than English. Section 7 briefly outlines the future of LCP research.

2 TYPES OF COMPLEXITY PREDICTION

2.1 Comparative Complexity

LCP originated as a sub-task of lexical simplification (LS) [127]. LS aims to replace complex words and expressions with simpler alternatives to improve the readability of a given text [95]. LCP is used by a LS system for two purposes: (1) to identify complex words, and (2) to rank the complexity of simplified word candidates.

Devlin and Tait [43] and Carroll et al. [27] adopted an LCP precursor within their LS systems' pipelines. They used WordNet [89] that applied Kučera-Francis's frequency norms to rank their synonymous and simplified word candidates on what they believed to be their level of simplicity. By doing so, their LCP sub-systems provided the most appropriate simplifications of their target complex words.

Using LCP to rank words in terms of their complexity gives rise to a unique type of complexity prediction: comparative complexity. Instead of determining whether a target word is either complex, non-complex, or a target word's level of perceived complexity, comparative complexity alternatively provides a value that distinguishes 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 [123, 127].

LS–2012 [127] is arguably the first shared-task that contained an LCP element, in the form of comparative complexity prediction. It tasked five participating teams to design systems to "rank a set of [candidate] words, from the simplest to the most difficult" [123]. Participating teams took into consideration a variety of features to conduct comparative complexity prediction. The most common of these features being simplified word candidates' frequency [10, 80, 123], n-grams [80, 123], morpho-syntactic characteristics including context [10, 65], and psycholinguistic properties [65].

2.2 Binary Complexity

Prior to 2018, complexity prediction research primarily focused on comparative as well as binary complexity prediction. Binary complexity prediction is referred to as complex word identification (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. Shardlow [115] experimented with a support vector machine (SVM) for CWI. He details the construction of a binary CWI dataset (Section 6.1.1) as well as the impact several features had on his CWI system's performance (Section 4).

CWI–2016 [95] was the first shared-task that challenged teams directly with binary CWI as a stand-alone task. This shared-task increased the popularity of complexity prediction research. CWI–2016 [95] is described in more detail within Section 3.1.

CWI's modeling as a binary classification task presented a few shortcomings during CWI–2016 [95]. The most notable is that CWI systems are unable to accurately and consistently classify target words on the decision boundary [157]. Words on the decision boundary are those words that have been labeled as being neither complex or non-complex but rather with an uncertain level of 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 [82, 157]. It is this disagreement in whether a word is either complex or non-complex during the annotation process, that creates target words with an uncertain level of complexity. As an example, within Maddela and Xu [82]'s Word Complexity Lexicon (Section 6.1.3) the following target words: "*substrate*" and "*derivatives*", were given non-complex labels

by 6 annotators and complex labels by 5 annotators. By averaging these labels, we can conclude that these words have an uncertain level of complexity. For a binary CWI system, such words, or word's on the decision boundary, are problematic. In this instance, "*substrate*" and "*derivatives*", would be labeled by a CWI system with a standard complexity threshold, as being non-complex given that this is their majority label. Nevertheless, this could be considered an inaccurate classification as these words may appear, to some, to be complex. Therefore, low inter-agreement leads to words on the decision boundary with an uncertain level of complexity. Being trained on such uncertain data often results in CWI systems misclassifying unseen target words. This, in turn, hinders their overall performance [117, 157].

2.3 Continuous Complexity

LCP was introduced to deal with words on the decision boundary [117]. LCP alternatively provides a complexity value that goes beyond purely assigning a complex or non-complex label. It models complexity on a continuum with varying degrees of difficulty. It assigns target words with a complexity label ranging from very easy to very hard. These labels are linked to certain thresholds, hence complexity values between 0 and 1: very easy (0), easy (0.25), neutral (0.5), difficult (0.75), or very difficult (1). Thus, by modeling complexity on a spectrum, LCP provides a more fine-grained representation of the complexity of a target word [117]. An LCP system is, therefore, better suited to deal with words on the decision boundary, such as neutral words, than compared to CWI systems.

LCP was not the first to treat lexical complexity as a continuous variable. Probabilistic complexity prediction was also a regression based task. However, different from LCP, probabilistic complexity prediction used continuous complexity values to make binary predictions [151]. This meant that its continuous complexity values were not used to distinguish varying degrees of difficulty, as is the case with LCP, but rather to indicate the probability of that target word being either complex or non-complex. CWI–2018 [151] developed systems for binary as well as probabilistic complexity prediction. It was the first shared-task that moved away from binary CWI. As such, it laid the foundations for what is known as LCP. CWI–2018 [151] is described in more detail within Section 3.2.

2.4 Personalized Complexity

Lee and Yeung [77] were interested in personalizing lexical simplification. They realized that prior CWI and LCP systems were unable to account for "variations in vocabulary knowledge among their users" [77], including other forms of idiosyncrasies, such as cross-linguistic influence. 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 [148]. As such, Lee and Yeung [77] believed that the previous "onesize-fits-all" approach to LCP was flawed and incapable of producing representative complexity predictions for a general populace. Instead, they introduced a new form of complexity prediction: personalized CWI. This approach creates personalized CWI systems that cater for 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 level [77, 159]. The Personalized LS Dataset is provided in Section 6.1.4.

2.5 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 [118, 151]. However, there exists little research into the complexity prediction of MWEs. Despite this, 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. [54]

provides "*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" [54], such as "*ballot filling*" or "*vote stuffing*". For this reason, the CompLex dataset [117] provides 1800 MWEs with preassigned complexity values. LCP–2021 [118], 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 [118] is described in greater detail with Section 3.3.

3 INTERNATIONAL COMPETITIONS

LCP has been the focus of several international competitions. Originating as a sub-task of LS–2012 [127] to a standalone task in CWI–2016 [95], LCP has become increasingly popular. In CWI–2018 [151], continuous complexity came into existence. Moving away from binary CWI, CWI–2018 [151] laid the foundations for LCP. LCP–2021 [118] then challenged teams to predict continuous complexity values of single words and MWEs. These shared–tasks have been described throughout the following sections (Sections 3.1, 3.2, and 3.3). Further detail regarding the architecture, development, and evolution of the systems submitted to these shared–tasks, is then provided in Section 4.

3.1 CWI–2016 at SemEval

The first CWI shared-task, referred to as CWI–2016, was organized at the International Workshop on Semantic Evaluation (SemEval).[3] CWI–2016 (SemEval–2016 Task 11) was modelled 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 [115], the LexMTurk Corpus [61], and Simple Wikipedia [37].

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 dataset. The training dataset included 2,237 target words in 200 sentences each annotated by 20 annotators. A word was considered complex in the training dataset if at least one of the 20 annotators assigned it as such. The test dataset 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 realistic scenario where the goal was to predict the individual needs of a speaker based on the needs of the target group [95]. 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 1 and discussed further in Section 4.

As can be seen in Table 1, 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-3 systems in CWI–2016, being PLUJAGH [145], LTG [84], and MAZA [85], also relied on probabilistic classifiers and on the aforementioned features. The F-scores achieved by the top-3 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. [157]. According to Zampieri et al. [157], this indicates 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.

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

Team	Classifiers	Features	Paper
AI-KU	SVM	word embeddings of the target and surrounding words	[74]
Amrita-CEN	SVM	word embeddings and various semantic and morphological features	[126]
BHASHA	SVM, Decision Tree	lexical and morphological features	[30]
ClacEDLK	Random Forests	semantic, morphological, and psycholinguistic features	[38]
CoastalCPH	Neural Network, Logistic Regression	word frequencies and word embeddings	[19]
HMC	Decision Tree and Random Forest	lexical, semantic, syntactic and psycholinguistic features	[105]
IIIT	Nearest Centroid	semantic and morphological features	[98]
JUNLP	Random Forest, naïve Bayes	semantic, lexicon-based, morphological and syntactic features	[92]
LTG	Decision Tree	n-grams and word length	[84]
MACSAAR	Random Forest, SVM	Zipfian frequency distribution, word length	[158]
MAZA	Meta-classifier	n-grams, word probability, word length	[85]
Melbourne	Weighted Random Forests	lexical and semantic features	[23]
PLUJAGH	Threshold-based methods	features extracted from Simple Wikipedia	[145]
Pomona	Threshold-based methods	word frequencies	[68]
Sensible	Ensemble Recurrent Neural Networks	word embeddings	[49]
SV000gg	System voting with threshold	morphological, lexical, and semantic features	[96]
TALN	Random Forest	lexical, morphological, semantic, and syntactic features	[110]
USAAR	Bayesian Ridge classifiers	hand-crafted word sense entropy metric and language model perplexity	[87]
UWB	Maximum Entropy	word occurrence counts on Wikipedia documents	[72]

Table 1. Systems submitted to the CWI–2016 in alphabetical order as summarized by [119].

3.2 CWI–2018 at BEA

The second edition of the CWI shared-task,[4] 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 four tracks were available, namely English, German, and Spanish monolingual. Furthermore, training and testing data from the multi-domain *CWIG3G2* dataset [152] was available for each language. The fourth track was the French multilingual track where only a French test dataset was available and the participants had to use the data made available for the other three languages to make predictions in French (Section 6.2.5)[5].

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 MWEs, and (3) two sub-tasks, one modelled as a binary classification task, and one modelled as a probabilistic classification task (Section 2).

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 2, 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 4).

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

[5] See Section 6.2.5 for more information regarding this type of complexity prediction: cross-lingual LCP.

Team	Classifiers	Features	Paper
Camb	Adaboost	N-grams, WordNet features, POS tags, dependency parsing relations, psycholinguistic features.	[52]
CFILT_IITB	Voting ensemble	Word length, syllable counts, vowel counts, WordNet-based features.	[141]
hu-berlin	naïve Bayes	Character n-grams	[103]
ITEC	LSTM	Word length, word and character embeddings, frequency count, psycholinguistics features.	[40]
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.	[59]
NLP-CIC	Tree Ensembles and CNNs	Word frequency, syntactic and lexical features, psycholinguistic features, and word embeddings.	[13]
SB@GU	Extra Trees	Word length, number of syllables, n-grams, frequency distribution.	[5]
TMU	Random Forest	Word length, word frequency, probability features derived from corpora.	[66]
UnibucKernel	Kernel-based learning with SVMs.	Character n-grams, semantic features, and word embeddings.	[26]

Table 2. Systems submitted to the CWI–2018 English binary classification single word track in alphabetical order as summarized by [119].

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 [151], the performance obtained by all teams on the News domain was generally substantially higher than the performance obtained in the two other domains.

3.3 LCP 2021 at SemEval

The 2021 Lexical Complexity Prediction Task [118], referred to as LCP–2021, was also held at SemEval and attracted 58 teams across its 2 sub-tasks. The shared-task introduced the term lexical complexity prediction (LCP), distinguishing its continuous annotations from the binary annotations of previous CWI tasks as described in Section 2.

The dataset [117] was developed using crowd sourcing. 10,800 instances were selected from three corpora covering the Bible [31], biomedical articles [71] and europarl [15]. 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 datasets, 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 [117]. 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

Team	Classifiers	Features	Paper
Alejandro Mosquera	Gradient Boosted Regression	Length, Frequency, Semantic, Sentence	[91]
Andi	Ridge Regression, Gradient Boosted Regression	Psycholinguistic, Glove, Word2Vec, ConceptNet NumberBatch, BERT, RoBERTa, ELECTRA, ALBERT, DeBERTa	[111]
Archer	Random Forest Regression, Gradient Boosted Regression	Length, Frequency, Psycholinguistic, Scrabble Score, Word Inclusion, Semantic	[114]
BigGreen	Gradient Boosted Regression, BERT	Length, Semantic, Glove, Elmo, InferSent, Phonetic, Frequency, POS	[64]
C3SL	Multi-layer Perceptron	Sent2Vec	[8]
Cambridge	BERT, Random Forest Regression	Frequency, Syntactic, Length	[153]
CLULEX	Decision Tree	Frequency, POS, Named Entities, Word Inclusion, Sentence, Bert	[125]
CompNA	Decision Tree Ensemble	Length, Semantic, Glove, Word Inclusion,	[136]
CS-UM6P	BERT, RoBERTa	Token and Context Encoded	[86]
CSECU-DSG	BERT, RoBERTa	Token and Context Encoded	[14]
DeepBlueAI	BERT, ALBERT, RoBERTa, ERNIE	Token and Context Encoded	[99]
IA PUCP	Gradient Boosted Regression	Sentence, POS, N-gram Frequency, RoBERTa, XLNet, BERT	[109]
IITK@LCP	Linear Regression, Support Vector Machine	Electra + Glove	[121]
JCT	Gradient Boosted Regression	POS, Frequency, BERT, Cluster Features	[79]
JUST-BLUE	Average of Weighted Bert and Roberta	Token Encoded and Context Encoded	[149]
Katildakat	Linear Regression, Multi-layer Perceptron	BERT, Length, BERT-score, Frequency, Semantic,	[137]
LAST	Gradient Boosted Regression	Frequency, Psycholinguistic, Sentence, Bigram Association	[17]
LCP-RIT	Random Forest Regressor	Length, Frequency, Character N-Grams, Psycholinguistic, POS	[41]
LRL_NC	Random Forest Regressor	Frequency, Semantic, Language Model, Psycholinguistic, Word Inclusion	[3]
Hub	RoBERTa, Inception	TF-IDF, Context Encoded	[63]
Manchester Metropolitan	CNN	Frequency, Psycholinguistic, Length, Embeddings	[47]
OCHADAI-KYOTO	BERT, RoBERTa	Token and Context Encoded	[133]
PolyU CBS-Comp	Gradient Boosted Regression	Frequency, Length, Capitalisation, POS, Embeddings, BERT, GPT-2	[147]
RG PA	RoBERTa	Context Encoded	[106]
RS_GV	Feed-Forward Neural Network	GLoVE, ELMo, BERT, Flair, Readability, Length, Frequency, Semantic, Psycholinguistic, Morphological, Word Inclusion, Named Entity	[130]
Stanford MLab	Gradient Boosted Regression	Glove, Length, POS, Named Entity	[112]
TUDA-CCL	Gradient Boosted Regression	Linguistic, Semantic, Embeddings, Psycholinguistic, Frequencies, Word Inclusion	[51]
UNBNLP	Neural Network, Support Vector Machine	Length, Frequency, Character-Level-Encoder, BERT	[70]
UPB	BERT, RoBERTa, Linear Regression	Transformers, Word Embeddings, Character Wmbeddings, Length, Psycholinguistic	[155]
UTFPR	Support Vector Machine	Frequency, Length, Semantic, Bert Embedding	[97]

Table 3. Systems submitted to LCP–2021 in alphabetical order as shown in [118].

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 datasets.

Several of the top-ranking systems for LCP–2021's sub-task 1 used transformer-based models [135]. However, systems that used hand-crafted features [91, 136, 147] also performed well with the top performing system [149]

in this category having achieved third place on the official ranking table. This is discussed further within Section 4.4.

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 [97, 121, 147] or predictions [91] 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.

4 APPROACHES TO PREDICTING LEXICAL COMPLEXITY IN ENGLISH TEXTS

Various models have been used for LCP. These range from support vector machines (SVMs), Decision Trees (DTs), Random Forests (RFs), neural networks to state-of-the-art transformers, such as Bert [42], RoBertA [81] and Electra [32]. 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 [95, 151]. 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.

4.1 Machine Learning Classifiers

4.1.1 **Support Vector Machines.** Support Vector Machines (SVMs) are statistical classifiers. They use labeled training data and engineered features to predict the class of unseen inputs [36, 115]. 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. For instance, it increases the probability of multiple features belonging to multiple classes. This subsequently decreases the decision margin between classes which in turn increases the likelihood of misclassification [30]. SVMs were popular within early LCP research since early research consisted entirely of comparative or binary complexity prediction: CWI [65, 127].

Jauhar and Specia [65] were one of the first to adopt an SVM for comparative 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 neighbouring 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 [115] created a complex word (CW) corpus consisting of 731 complex words in context [116] (Section 6). 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 2).

Kuru [74] was interested in the use of Glove word embeddings [100] for capturing the contextual information of a target word. Building on Jauhar and Specia [65]'s bag-of-words approach in extracting contextual information, Kuru [74] 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, which they submitted to CWI–2016 [95]. The first model: AIKU (native), was trained on the "word embedding of the target word and its substrings as features" [74]. 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 neighbouring words. They discovered that both of their models performed equally well having attained matching G-scores of 0.545 at CWI–2016 [95]. This led Kuru [74] to conclude that contextual information, such as a target word's neighbouring words, is not a useful feature in improving the CWI performance of a SVM model.

S.P et al. [126] experimented with Word2vec word embeddings alongside statistical, POS-tag, and similarity features. They trained four SVM models. Their first model was trained only 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 neighbouring 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 F-score of 0.109 and a G-score of 0.547 [95, 126]. It comes as no surprise that given their reliance on word-embeddings and contextual information, S.P et al. [126]'s AmritaCEN (w2vecSim) and Kuru [74]'s AIKU (native1) have both achieved similar performances. However, an interesting observation is that S.P et al. [126]'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 [6].

4.1.2 **Decision Trees.** Decision trees (DTs) make predictions based on a set of learnt 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 [95]. This may be due to DTs being better suited in dealing with features that overlap between classes, given their reliance on learnt rules rather than prototypical features, such as support vectors.

Throughout CWI–2016, as detailed in Section 3.1, the most common and arguably the most successful CWI systems consisted of either a DT or a Random Forest (RF) model [95]. This marked LCP's transition to DTs and RFs. These models maintained state-of-the-art status until LCP–2021 [118]. 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 [115]'s, Jauhar and Specia [65]'s, 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 4.2.

Choubey and Pateria [30] investigated the performance of both a SVM and a DT at CWI. They discovered that their "SVM seemed to be less effective for CWI" [30, 95]. Their SVM attained a F-score of 0.179 and a G-score of 0.508, whereas their DT produced a F-score of 0.181 and a G-score of 0.529 [30]. They reasoned that their SVM's worst performance was due to it having "overlapping decision boundaries" [30]. Again, this refers to the decision boundary misclassification problem that is commonly faced by CWI systems (Section 2).

The systems submitted by Quijada and Medero [105], referred to as HMC, were among the top performing systems at CWI–2016 [95, 105]. One of their systems consisted of a DT, known as HMC (DecisionTree25), whereas the other consisted of a regression tree (RT), called HMC (RegressionTree05) [105]. These models outperformed their SVM counterpart, with the DT model achieving a F-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 learnt. These rules were learnt from several inputted features. These features belonged to two main categories: statistical, and psycholinguistic [7]. 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 [105]. Their psycholinguistic features included age-of-acquisition, perceived word concreteness and the number of differing pronunciations

[6]The poor performance of POS-tags as a feature for LCP was recently verified by Desai et al. [41].
[7]HMC also utilized POS-tags as features.

associated with a target word. They claimed that their models' success was due to their use of corpus-based features, especially their use of unigram and lemma frequencies.

4.1.3 **Random Forests.** 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.

Ronzano et al. [110] submitted a RF to CWI–2016 that outperformed other DT models [95]. Their RF, referred to as TALN (RandomForest_WEI), was taken from the Weka machine learning framework [57]. 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 [105]. However, like other models submitted to CWI–2016, additional features were also exploited, such as contextual features [95]. 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) [34, 78] and the 2014 English Wikipedia Corpus [35] [8]. 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 [105]. TALN (RandomForest_WEI) achieved an F-score of 0.268 and a G-score of 0.772. This was a respective -0.02 less than the F-score and +0.006 better than the G-score achieved by the best performing HMC system [95, 110].

Zampieri et al. [158] 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 [105, 158]. 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 a F-score of 0.270 and a G-score of 0.754 at CWI–2016 [95] giving it a greater F-score of +0.002, yet an inferior G-score of -0.018 than compared to TALN [110]. Per their model's performance, Zampieri et al. [158] concluded that Zipfian features are good baseline indicators of lexical complexity.

Davoodi and Kosseim [38] experimented with several models for CWI–2016 [95]. 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 [38] had a particular interested in psycholinguistic features, namely abstractness. They believed there existed a correlation between "the degree of abstractness of a word and its perceived complexity"[9] [38]. They developed two RF models. Their first RF had a threshold of 0.5, whereas their second had a threshold of 0.6. This meant that for a target word to be classified as being complex, their 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 outperformed their first by a G-score of +0.028. As such, having a higher threshold for complexity assignment would appear to improve CWI performance.

[8]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.

[9]Non-complex words are theorised to have more concrete meanings than complex words, hence complex words are believed to be more abstract in regards to their meaning [25].

4.2 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, whilst 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 [95] to CWI–2018 [151], 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 4.3.1).

Malmasi and Zampieri [85] 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" [85]. These base classifiers then input their predicts 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" [85].

Malmasi and Zampieri [85] submitted two ensemble-based models to CWI–2016: MAZA A and MAZA B [95]. Both of these models' base classifiers were decision stumps. Decision stumps 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 a 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"[85].

MAZA B was trained using additional contextual features that were not utilized by MAZA A [85]. 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 neighbouring 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 a F-score of +0.116 greater than MAZA A [85, 95]. Malmasi and Zampieri [85] contribute 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[10].

Choubey and Pateria [30] constructed two ensemble-based models for CWI–2016 [95]. The first, referred to as GARUDA (HSVM&DT), had a meta-classifier architecture which 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 [30]'s second ensemble-based model contained twenty 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 dataset. 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 worst than individual SVM or DT models, whereas their second ensemble-based model achieved average performance. They blamed this poor performance on the "overlapping

[10]Malmasi and Zampieri [85] would appear to contradict Kuru [74], as Kuru [74] 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 [149] (Section 4.3.1).

decision boundaries" [30] of their SVM sub-models. This once again demonstrates the inferiority of SVMs for CWI.

The SV000gg systems, created by Paetzold and Specia [95], were the best performing systems submitted to CWI–2016 [95, 96]. Paetzold and Specia [95] adopted an ensemble-based model that utilized a variety of models. They believed that model diversity would result in greater CWI performance. Their ensemble-based model consisted of a lexicon-based model, a threshold-based model to SVMs, DTs, RFs and other machine learning 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.

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 regards 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-orientated soft voting determines the final label of a target word by examining the performances of each sub-model "over a certain validation dataset such as precision, recall, and accuracy" [96]. 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 F-score of 0.246 and a G-score of 0.774. Hard SV000gg attained a slightly worst F-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 regards to its G-score, including those mentioned above [95]. As a result, both models demonstrated the superiority of diverse ensemble-based models for binary CWI in comparison to other models.

Gooding and Kochmar [52] were inspired by the performance of prior ensemble-based models at CWI–2016 [95]. Their system, referred to as Camb, achieved good results on both of CWI–2018's sub-tasks: binary CWI, and probabilistic complexity prediction (Section 3) [151]. Camb used a boosting classifier: AdaBoost, with 5000 estimators followed by a RF bootstrap aggregation model [52]. They experimented with differing sub-models, each being trained on a set of given features similar to those used by prior CWI systems [95]. They concluded that an ensemble-based model that combines both AdaBoost and a RF with equal weights, consistently produced the best performance [52].

Aroyehun et al. [13] experimented with the tree learner model provided by KNIME [16], along with other combinations of DTs, RFs, and gradient boosted tree learners for CWI–2018's sub-task 2: probabilistic complexity prediction [13, 95]. 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 3). Therefore, Gooding and Kochmar [52] and Aroyehun et al. [13] demonstrated that ensemble-based models achieve good performance at binary as well as probabilistic complexity prediction.

4.3 Neural Networks

Deep learning is highly popular within NLP and Computational Linguistics having achieved state-of-the-art performance in various NLP-related tasks [44, 146]. Neural networks attempt to mimic human learning. They achieve this by manipulating weight values between nodes that contain characteristic information, or learnt features, related to the input. 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 [95, 151]. It was not until the introduction of probabilistic complexity prediction (Section 2), that some neural networks were shown to perform well, and on occasion, on par with more traditional models [13, 151].

Gillin [49] was one of the first to investigate the performance of a recurrent neural network (RNN) at 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 overtime, thus, inhibiting the weight values of earlier nodes from being accurately updated [49, 50]. This impairs a neural network's ability to retain information learnt at earlier stages. A GRU counters this problem by acting as a "memory" device [49]. It controls what new information should be learnt, what prior information should be remembered, and what previous information should be forgotten, when updating a weight value.

Gillin [49] submitted their RNN model with a GRU as well as a 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 [49, 95]. Nevertheless, in comparison to other more traditional models, Sensible (baseline) performed poorly. It attained an F-score of 0.140 and a G-score of 0.646. Gillin [49] claimed it was the small size of CWI-2016's training dataset that caused their RNN model to perform less well than expected (Section 3.1).

Aroyehun et al. [13] were the first to experiment with a convolutional neural network (CNN) for 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 transformers said input before passing it onto a further layer. However, CNN models lack the temporal capabilities of an RNN with an embedded GRU. Regardless of this limitation, Aroyehun et al. [13]'s CNN, 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 [151] (Section 4.2). 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 [59] compared models that adopted feature engineering to neural networks at CWI–2018 [151]. 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 1 & 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 a additive gradient structure; being parallel to the use of a GRU.

For binary CWI [151], Hartmann and dos Santos [59]'s feature engineered XGBoost model outperformed their neural network models. It attained an F-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 a F-score of 0.588. Their feature engineered XGBoost model, and their shallow neural

network model, achieved less impressive F-scores of 0.2978 and 0.2958 respectively. Both Aroyehun et al. [13] and Hartmann and dos Santos [59], therefore, proved the viability of neural networks for probabilistic complexity prediction.

4.3.1 **Transformers.** The best performing systems of LCP–2021 [118] used transformer-based models. Transformer-based models were introduced to overcome the limitations associated with prior neural networks, such as RNNs, LSTM, and CNN models [13, 49, 59]. The primary limitation of these models being the vanishing gradient problem [50].

The models introduced by Hartmann and dos Santos [59] and Aroyehun et al. [13] attempted to counter this problem by having a GRU, forget gate, or an additive gradient structure. However, these solutions did not completely resolve the vanishing gradient problem but rather reduced its severity [62].

Transformer-based models, such as Bert [42], RoBertA [81], and Electra [32], do not suffer from gradient vanishing. Instead, they rely purely on an attention mechanism that allows them to capture long-term dependencies within a dataset more effectively than prior neural networks [50]. This may explain the superior performance of transformer-based models at LCP [118].

Just Blue by Yaseen et al. [149], achieved the highest Pearson's Correlation at LCP–2021's sub-task 1 of 0.7886 [118]. 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 [149].

Just Blue consisted of an ensemble of BERT [42] and RoBERTa [81] transformers. It 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. Howbeit, 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 [85, 105]. 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. [149] experimented with different models as well as different weight splits between their target word and sentence level inputs. They discovered that between a SVM, RF, BERT, RoBERTa, or a BERT and RoBERTa, a BERT and RoBERTa ensemble-based 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.

Pan et al. [99]'s system, referred to as DeepBlueAI, achieved second place at LCP–2021's sub-task 1 and first place at sub-task 2 [99, 118]. 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 [42], RoBERTa [81], ALBERT [75], and ERNIE [131]. 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 7-fold cross-validation to check for overfitting or selection bias, with the fourth layer then having adopted training strategies, such as data augmentation and pseudo-labelling. Data augmentation is the training strategy of adding new data to a training dataset 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" [99, 143]. Pseudo-labelling is the training strategy of predicting labels for unlabeled data and then adding the newly labeled data back into the training dataset. 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 (See Equation 1):

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

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. [99] contributed 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 regards to Just Blue's high performance [149], it would appear that current state-of-the-art LCP systems consist of an ensemble of differing transformers-based models.

RG_PA, created by Rao et al. [106], was the second highest performing system at LCP–2021's sub-task 2 having achieved a Pearson's Correlation of 0.8575 [106, 118]. Unlike Just Blue [149] and DeepBlueAI [99], 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 [11]. 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 dataset. 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 dataset. 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.

4.4 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 [91]. Mosquera [91] approached sub-task 1 from a more traditional feature engineering approach. Much like prior CWI sytems, Mosquera [91] utilized a combinations of lexical, contextual, and semantic features (Section 2.2). However, unlike previous CWI systems, these features were extensive with 51 features in total being used to rate lexical complexity. Many of these features have never before been exploited for LCP. These features include SUBTLEX features, word etymology, and several readability indices. SUBTLEX features are those features that were 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 [24]. 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 [46], Gunning-Fog index [56], LIX score [11], SMOG index [88], and Dale-Chall index [28].

Mosquera [91] 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 [82]'s word complexity lexicon. The importance of age and prior complexity labels was likewise verified by Desai et al. [41]. Mosquera [91] also

[11] 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 dataset.

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 the perceived comprehension difficulty associated with a sentence. It is likely that Mosquera [91]'s extensive list of features, inclusion of such contextual features, or contextual readability measures, along with their use of a LGB model, is responsible for their system outperforming a similar feature engineering approach by Desai et al. [41] [118].

4.5 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. [149] and Pan et al. [99] demonstrate the importance of model diversity with an ensemble-based model. They show that a combination of differing transformers produces the best results for LCP in comparison to the use of multiple transformers of the same type. (2) Transformer-based models are less prone to gradient vanishing. Therefore, their context vector representations, which contain contextual information learnt from the entire dataset, have been found to enhance LCP performance. This being beyond that potentially achievable by prior neural networks that are prone to gradient vanishing, and thus the depreciation of their previously learnt input (Section 4.3). As such, ensembles-based models that rely on multiple transformers of varying types and that take into consideration contextual information of the target word, are currently the most state-of-the-art systems for LCP. However, Mosquera [91] 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.

5 USE CASES AND APPLICATIONS

LCP has many potential use cases and applications [95, 118, 151]. LCP systems can be utilized within a variety of assistive technologies, such as intelligent tutoring systems (ITSs) or computer-assisted language learning (CALL) applications. They can also aid other downstream NLP-related tasks. These tasks include sentiment analysis (Section 5.3.1), authorship identification (Section 5.3.2), and machine translation (Section 5.3.3).

5.1 Improving Readability

ITSs are "computer learning environments designed to help students master difficult knowledge and skills" [55]. 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. CALL applications are subsequently ITSs that specialize in language learning and have been found to improve second language (L2) acquisition [134]. These applications include multiple designs, are based on differing pedagogical practices, and allow for varying degrees of learner-computer interaction [7].

A common approach among CALL is to simplify a text so as to make it more accessible for the L2 learner [108]. Morris, et al., 2013 [156]. Alhawiti [6] states that TS can be beneficial to language learners. Therefore, an ITS or CALL application that incorporated TS would likewise be beneficial. For instance, TS is believed to significantly increase the literacy [101] as well as advance the vocabulary development of L2 learners [108, 132].

Rets and Rogaten [108] tested 37 participants on their ability to memorize and process the ideas presented within two texts: an authentic text, and 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 [108]) to conclude that TS leads to better textual comprehension which correlates with a greater learning potential [104, 108].

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” [122]. 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 [27, 107]. ITSs developed for children may likewise use TS in order to reduce the amount of high-level jargon, or non-frequent words, within a text [39].

TS is, therefore, extremely useful in improving the vocabulary and literacy development of L2 learners [6], people with autism [122], dyslexia [27, 107], or aphasia [27], as well as children [39]. However, to achieve TS, a number of preprocessing tasks need to be completed, such as LCP.

5.2 Text Simplification Pipeline

LCP is a precursor to LS, which in turn is a precursor to TS. LCP is subsequently a key component of the TS pipeline. For instance, as previously mentioned within Section 2.1, an LCP system can identify all of the complex words within a given text, and then either explain these complex words [9], input them into a readability calculator [60], or pass them onto a LS system [123]. A LS system[12] will then replace these complex words with simpler alternatives, such as substituting the complex word “*folly*” for the synonymous and less archaic “*foolishness*” [53, 82]. A TS system then uses the output from a LS system to provide a simplified text. This simplified text can then be used by the learner to better understand the meaning of the original text, or complex word. This is demonstrated the example below taken from the Complex dataset [117].

Original **Simplified**

"Folly is set in great dignity" = "Foolishness is in pride"

5.3 Emerging Use Cases

LCP can aid other downstream NLP-related tasks: sentiment analysis [29, 69], authorship identification [1, 128], and machine translation [138]. The benefits of incorporating LCP within these tasks have been hypothesized in the following sections (5.3.1 to 5.3.3).

5.3.1 **Sentiment Analysis.** Sentiment analysis (SA) deals with "subjective statements" [69]. It is the task of extracting and identifying opinions, attitudes, and emotions, expressed within a text [29, 69]. However, different types of sentences express sentiment in different ways. Non-complex sentences, such as: "*It is adequate.*", express sentiment clearly, whereas more complex sentences: "*It is adequate?*", may connote a more convoluted opinion. The second sentence, through its use of a question mark, becomes an interrogative sentence. This changes its meaning from a positive statement to a potentially cynical question [29]. The job of a LCP system within a SA pipeline would be to differentiate between those instances which depict a non-complex instance: "*adequate.*", with those which depict a complex instance: "*adequate?*". For example, an individual with low English proficiency may be more familiar with the first declarative instance of the word, yet less familiar with its second interrogative instance, hence assigning its second instance as being more complex. Such information may then be entered into an SA system as a feature or be used by an SA system to select the correct approach in extracting a target word’s meaning. As such, LCP may increase the quality of an SA system’s output [29].

5.3.2 **Authorship Identification.** Authorship identification is the task of identifying the author of a given text [20]. 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" [73]. The TTR, therefore, shows the diversity of a author’s vocabulary. LCP provides an

[12] A LS system aims to "replace complex words and expressions with simpler alternatives" [95].

additional measurement of vocabulary richness. Adding to the TTR, it provides an average lexical complexity fingerprint 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.

5.3.3 **Machine Translation.** Before TS shifted to focus on improving the readability of texts for language learners, individuals with a reading disability, or another target demographic, TS's primary focus was to aid machine translation (MT) [4]. MT is the task of automatically translating a source language into a target language [138]. 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, 129, 138]. TS achieves this by reducing the ambiguity of the inputted texts in the source language [138]. 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. LCP systems that enhance the performance of TS may, therefore, aid MT.

6 RESOURCES

6.1 Additional English Datasets and Resources

The shared-tasks of CWI–2016 [95], CWI–2018 [151], and LCP–2021 [118], tested participating teams on three datasets that have since contributed significantly to LCP research (Section 3). Nevertheless, these are not the only influential datasets that have provided 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 4. Apart from the CWI–2016, CWI–2018, and the CompLex datasets already discussed within Section 3, the remaining datasets are introduced throughout the following sections (6.1.1 to 6.1.4) .

6.1.1 **CW Corpus.** The CW Corpus contains 731 complex words in context [116]. 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 [116] 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 differs 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.

6.1.2 **Middlebury Corpus.** Horn et al. [61] 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 US 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 [61]. Similar to Shardlow [116], 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 Horn et al. [61]'s corpus a valuable resource for investigating comparative complexity.

6.1.3 **Word Complexity Lexicon.** Maddela and Xu [82] recognized the limitations of prior CWI datasets, namely, the limitations associated with using binary complexity labels rather than continuous complexity values (Section 2) [82]. As a response to these limitations, they constructed the Word Complexity Lexicon (WCL). The

Dataset	Complexity	Size	Annotators	Comments	Paper
LS–2012	Comparative	201 complex words each with several candidate simplifications.	Native English speakers provided simplifications and 4 L2 learners ranked these simplifications based on their complexity.	Each Complex word is shown in 10 different contexts.	[127]
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.	[116]
Middlebury Corpus	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.	[61]
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 dataset included 2,237 target words in 200 sentences, whereas the test dataset included 88,221 target words in 9,000 sentences.	[95]
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, 7015, 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.	[151]
Word Complexity Lexicon	Continuous	15,000 words labeled with varying degrees of complexity.	11 non-native yet fluent English speakers.	Used a six-point likert scale for annotation.	[82]
Personalized LS Dataset	Personalized - Continuous	12,000 words labeled with varying degrees of complexity.	15 learners of English, who were native Japanese speakers.	"	[77]
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 five-point likert scale for annotation. Words in context were taken from the Bible, biomedical articles, and europarl.	[117]

Table 4. Datasets used for English complexity prediction research arranged in chronological order.

WCL is a dataset made up of "15,000 English words with word complexity values assessed by human annotators" [82]. These 15,000 words were the most frequent 15,000 words found within the Google 1T Ngram Corpus [21]. 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 was believed to be due to the differing characteristics of their native languages, hence caused by cross-linguistic influence[13].

[13]Cross-linguistic influence is defined as the effects a bilingual speaker's L1 has on their L2 production, hence complexity assignment [148]. See Section 2.4 for more details.

6.1.4 **Personalized LS Dataset.** Lee and Yeung [77] constructed a dataset of 12,000 words for personalized CWI. These words were ranked on a five-point likert scale. 15 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" [77]. Lee and Yeung [77] 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 [77] is an ideal resource for future personalized CWI or LCP research.

6.2 Lexical Complexity Prediction in Other Languages

Since CWI–2018 [151] (Section 3.2), LCP for other languages has began to receive more attention in the form of monolingual, multilingual, and cross-lingual LCP [45, 150]. 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 in one or multiple languages and then using that system to predict the lexical complexities of a language previously unseen within the training dataset.

6.2.1 **Chinese.** Lee and Yeung [76] created a SVM designed to identify Chinese complex words. Their monolingual LCP model was then further developed by Yeung and Lee [150]. 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 [76]. These being, the target word's ranking in a Chinese proficiency test[14], word length, word frequency in the Chinese Wikipedia Corpus [76] and the Jinan Corpus of Learner Chinese (JCLC) [140], along with Chinese character frequency in the JCLC. They discovered that their logistic regression models outperformed Lee and Yeung [76]'s prior SVM. They also found that their model was better at predicting the lexical complexities of their annotators with low Chinese L2 proficiency than compared to those who had high Chinese L2 proficiency.

6.2.2 **Japanese.** Nishihara and Kajiwara [93] 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 thousand Japanese words divided into three levels of difficulty: easy, medium, or difficult. Nishihara and Kajiwara [93] also rated the complexity of words from Japanese Wikipedia, the Tsukuba Web Corpus [94], and the Corpus of Contemporary Written Japanese [83]. 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" [93]. 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

[14]The Hanyu Shuiping Kaoshi (HSK) [58].

indicator of a word's complexity. Nishihara and Kajiwara [93] concluded that the use of such language specific features was responsible for their model's good performance.

6.2.3 **Swedish.** Smolenska [124] 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 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. Whilst evaluating the quality of the two datasets, Smolenska [124] discovered that the second dataset correlated better with the judgements of two human evaluators. Results showed that the RF model achieved the best performance having been trained on a number of features, including morpho-syntactic, contextual, conceptual, and frequency based features.

6.2.4 **Multilingual LCP.** Sheang [120] saw the advantages of adopting a feature engineering approach as well as a CNN model for multilingual CWI. As a result, Sheang [120] 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 [151] (Section 3.2), this multilingual model was found to outperform the best performing model of CWI–2018 [151] on the Spanish and German datasets.

Aprosio et al. [12] created a LCP system that caters for the native language of the user. As previously discussed within Section 2.4, an annotator's or user's native language influences their perception of lexical complexity through what is known as cross-linguistic influence. As such, Aprosio et al. [12]'s 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" [12]. 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 [22], Normalized Edit Distance [139], and Jaro/Winkler [144]. Once potential cognates had been identified, their system uses an SVM to classify which of these cognates may, in fact, be false friends. Their SVM is 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. [12] created a LCP system that can recognize and exploit language-dependent features to improve its performance. Aprosio et al. [12] is, therefore, a good example of personalized LCP.

6.2.5 **Cross-Lingual LCP.** Finnimore et al. [45] continued working on CWI–2018's sub-task 1[15] [151] (Section 3.2). 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 n-grams is an example of such a language-dependent feature, since n-grams denote the unique character combinations, hence spellings of a particular language. Instead, Finnimore et al. [45] 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 unigram

[15]CWI–2018: Sub-task 1 being binary complex prediction in multiple languages (Section 3.2).

probabilities associated with a target word[16]. 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 [18] provides further evidence in favor of cross-lingual learning for LCP. Their cross-lingual CWI system achieved the best F-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" [18]. 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. [154] experimented with several transformer-based models, such as Multilingual BERT (mBERT) [102] and XLM-RoBERTa [33], 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" [102]. XLM-RoBERTa is also pretrained on 100 languages, yet with more data [33]. Zaharia et al. [154] tested these models performance on the WikiNews datasets provided by CWI–2018 [151]. They found that XLM-RoBERTa was the best performing model. It achieved a higher F-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 [33, 154].

7 SUMMARY AND OUTLOOK

This paper presented an overview of computational models applied to lexical complexity prediction with a specific focus on English. We carried out a comprehensive survey of related work addressing the different types of computational modelling applied to this problem, such as comparative, binary, continuous, and personalized complexity. We discussed the international shared-tasks organized to serve as performance benchmarks (Section 3): CWI–2016 [95], CWI–2018 [151], and LCP–2021 [118]. We explained the architecture, development, and evolution of LCP systems, ranging from feature engineering approaches, neural networks to the most recent state-of-the-art transformer-based models (Section 4). We presented various use cases and applications of lexical complexity prediction, including for other NLP-related tasks such as sentiment analysis (Section 5.3.1), author identification (Section 5.3.2), and machine translation (Section 5.3.3). We collected and summarized English datasets used for LCP (Section 6.1) and also briefly presented work on languages other than English (Section 6.2): Chinese [76], Japanese [93], and Swedish [124].

There now exists an unprecedented demand for LCP research. With distance learning becoming ever more popular and with LCP now acting as a precursor within other NLP-related tasks, the future for LCP research would appear to be promising. LCP–2021 [118] 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 [99, 149]. CWI–2018 along with others [18, 45, 154], have proven that cross-lingual LCP is viable. LCP is now being conducted for languages other than English [76, 93, 124]. 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. We expect such personalized LCP systems to become popular as their datasets are likely to contain more consistent complexity ratings due to

[16]Finnimore et al. [45] did not take into consideration cognates which are also transferable as proven by Aprosio et al. [12].

there being less disagreement among their annotators. Hence, such personalized LCP systems may achieve new levels of state-of-the-art performance.

ACKNOWLEDGMENTS

The authors would like to thank Richard Evans for the valuable suggestions and feedback provided.

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, New Orleans, United States.

[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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).

[4] Suha S. Al-Thanyyan and Aqil M. Azmi. 2021. Automated Text Simplification: A Survey. *ACM Comput. Surv.* 54, 2, Article 43 (March 2021), 36 pages.

[5] 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, New Orleans, United States.

[6] Khaled M. Alhawiti. 2015. Innovative Processes in Computer Assisted Language Learning. *International Journal of Advanced Research in Artificial Intelligence* 4, 2 (2015), 7–13.

[7] Ali Alkhatlan and Jugal Kalita. 2018. Intelligent Tutoring Systems: A Comprehensive Historical Survey with Recent Developments. arXiv:1812.09628 [cs.HC]

[8] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).

[9] Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics* 46, 1 (2020), 135–187.

[10] Marilisa Amoia and Massimo Romanelli. 2012. SB: mmSystem - Using Decompositional Semantics for Lexical Simplification. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Association for Computational Linguistics, Montréal, Canada, 482–486.

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

[12] 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, Genoa, Italy, 192–200.

[13] 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, New Orleans, United States.

[14] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).

[15] 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.

[16] 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. 11, 1 (2009).

[17] Yves Bestgen. 2021. LAST at SemEval-2021 Task 1: Improving Multi-Word Complexity Prediction Using Bigram Association Measures. In *Proceedings of the Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).

[18] 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, New Orleans, United States.

[19] Joachim Bingel, Natalie Schluter, and Héctor Martínez Alonso. 2016. CoastalCPH at SemEval-2016 Task 11: The importance of designing your Neural Networks right. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, 1028–1033.

[20] Tudor Boran, Muhamet Martinaj, and Md Shafaeat Hossain. 2020. Authorship identification on limited samplings. *Computers & Security* 97 (2020), 101943.
[21] Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram version 1. *Linguistic Data Consortium (LDC)* (2006).
[22] Chris Brew and D. McKelvie. 1996. Word-Pair Extraction for Lexicography. In *Word-Pair Extraction for Lexicography*. 45–55.
[23] 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-2016)*. Association for Computational Linguistics, San Diego, California, 975–981.
[24] 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.
[25] 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.
[26] 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, New Orleans, United States.
[27] John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical Simplification of English Newspaper Text to Assist Aphasic Readers. In *Proceedings of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*. 7–10.
[28] Jeanne Sternlicht Chall and Edgar Dale. 1995. *Readability revisited: The new Dale-Chall readability formula*. Brookline Books.
[29] Tao Chen, Ruifeng Xu, Yulan He, and Xuan Wang. 2017. Improving sentiment anaylsis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems With Applications* 72, 1 (2017), 221–230.
[30] 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-2016)*. Association for Computational Linguistics, San Diego, California, 1006–1010.
[31] Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: the Bible in 100 languages. *Language Resources and Evaluation* 49, 2 (2015), 375–395.
[32] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555* (2020).
[33] 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, 8440–8451.
[34] British National Corpus. 2015. (2015). http://www.natcorp.ox.ac.uk
[35] The Wikipedia Corpus. 2001. (2001). https://www.english-corpora.org/wiki/
[36] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning* 20, 1 (1995), 273–297.
[37] 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, Portland, Oregon, USA, 665–669.
[38] 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-2016)*. Association for Computational Linguistics, San Diego, California, 982–985.
[39] Jan De Belder and Marie-Francine Moens. 2010. Text simplification for children. In *Proceedings of the SIGIR workshop on accessible search systems*. ACM, Geneva, 19–26.
[40] 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, New Orleans, United States.
[41] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).
[42] 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: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.
[43] 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.
[44] Ehsan Fathi and Babak Maleki Shoja. 2018. Chapter 9 - Deep Neural Networks for Natural Language Processing. In *Computational Analysis and Understanding of Natural Languages: Principles, Methods and Applications*. Handbook of Statistics, Vol. 38. Elsevier, 229–316.
[45] 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: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota.

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

[47] Robert Flynn and Matthew Shardlow. 2021. Manchester Metropolitan at SemEval-2021 Task 1: Convolutional Networks for Complex Word Identification. In *Proceedings of the Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).

[48] 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.

[49] 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-2016)*. Association for Computational Linguistics, San Diego, California, 963–968.

[50] 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, Sofia, Bulgaria, 179–183.

[51] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).

[52] 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, New Orleans, United States.

[53] Sian Gooding and Ekaterina Kochmar. 2019. Recursive Context-Aware Lexical Simplification. 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)*. 4855–4865.

[54] 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 (READI2020)*. European Language Resources Association, Marseille, France, 14–19.

[55] 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.

[56] Robert Gunning. 1952. *The technique of clear writing*. McGraw-Hill, New York.

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

[58] Hanban. 2014. *International curriculum for Chinese language education*. Beijing Language and Culture University Press, Beijing, China.

[59] 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, New Orleans, United States.

[60] Michael Heilman, Kevyn Collins, and Jamie Callan. 2007. Combining Lexical and Grammatical Features to Improve Readability Measures for First and Second Language Texts. In *Proceedings of NAACL HLT*. Association for Computational Linguistics, 460–467.

[61] 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, Baltimore, Maryland, 458–463.

[62] Yuhuang Hu, Adrian Huber, Jithendar Anumula, and Shih-Chii Liu. 2019. Overcoming the vanishing gradient problem in plain recurrent networks. arXiv:1801.06105 [cs.NE]

[63] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).

[64] Aadil Islam, Weicheng Ma, and Soroush Vosoughi. 2021. BigGreen at SemEval-2021 Task 1: Lexical Complexity Prediction with Assembly Models. In *Proceedings of the Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).

[65] Sujay Kumar Jauhar and Lucia Specia. 2012. UOW-SHEF: SimpLex – Lexical Simplicity Ranking based on Contextual and Psycholinguistic Features. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Association for Computational Linguistics, Montréal, Canada, 477–481.

[66] 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, New

Orleans, United States.
[67] 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 2013)*. 59–73.
[68] 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-2016)*. Association for Computational Linguistics, San Diego, California, 1047–1051.
[69] Muhammad Taimoor Khan, Mehr Durrani, Armughan Ali, Irum Inayat, Shehzad Khalid, and Kamran Habib Khan. 2016. Sentiment analysis and the complex natural language. *Complex Adaptive Systems Modeling* 4, 2 (2016), 1–19.
[70] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).
[71] Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*.
[72] 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-2016)*. Association for Computational Linguistics, San Diego, California, 1038–1041.
[73] Miroslav Kubát and Jiří Milička. 2013. Vocabulary Richness Measure in Genres. *Journal of Quantitative Linguistics* 20, 4 (2013), 339–349.
[74] 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-2016)*. Association for Computational Linguistics, San Diego, California, 1042–1046.
[75] 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. arXiv:1909.11942 [cs.CL]
[76] John Lee and Chak Yan Yeung. 2018. Automatic prediction of vocabulary knowledge for learners of Chinese as a foreign language. In *In Proceedings of the 2nd International Conference on Natural Language and Speech Processing (ICNLSP)*.
[77] John Lee and Chak Yan Yeung. 2018. Personalizing Lexical Simplification. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 224–232.
[78] Geoffrey Leech, Paul Rayson, et al. 2014. *Word frequencies in written and spoken English: Based on the British National Corpus*. Routledge.
[79] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).
[80] 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 First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Association for Computational Linguistics, Montréal, Canada, 487–492.
[81] 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. *arXiv preprint arXiv:1907.11692* (2019).
[82] 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*. 3749–3760.
[83] 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 Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA), Valletta, Malta.
[84] 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-2016)*. Association for Computational Linguistics, San Diego, California, 996–1000.
[85] 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-2016)*. Association for Computational Linguistics, San Diego, California, 991–995.
[86] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).
[87] 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-2016)*. Association for Computational Linguistics, San Diego, California, 958–962.
[88] Harry McLaughlin. 1969. SMOG grading – a new readability formula. *Journal of Reading* 22 (1969), 639–646.
[89] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[90] 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.
[91] Alejandro Mosquera. 2021. Alejandro Mosquera at SemEval-2021 Task 1: Exploring Sentence and Word Features for Lexical Complexity Prediction. In *Proceedings of the Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).
[92] 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-2016)*. Association for Computational Linguistics, San Diego, California, 986–990.
[93] 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, Marseille, France.
[94] University of Tsukuba. 2014. *NLT Corpus*. National Institute for Japanese Language and Linguistics, Lago Institute of Language: NINJAL-LWP for TWC.
[95] Gustavo Paetzold and Lucia Specia. 2016. SemEval 2016 Task 11: Complex Word Identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, 560–569.
[96] 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-2016)*. Association for Computational Linguistics, San Diego, California, 969–974.
[97] Gustavo Henrique Paetzold. 2021. UTFPR at SemEval-2021 Task 1: Complexity Prediction by Combining BERT Vectors and Classic Features. In *Proceedings of the Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).
[98] 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-2016)*. Association for Computational Linguistics, San Diego, California, 1017–1021.
[99] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).
[100] 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)*. 1532–1543.
[101] Sarah Petersen and Mari Ostendorf. 2009. A machine learning approach to reading level assessment. *Computer Speech & Language* 23, 1 (2009), 89 – 106.
[102] 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, Florence, Italy.
[103] 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, New Orleans, United States.
[104] 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.
[105] 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-2016)*. Association for Computational Linguistics, San Diego, California, 1034–1037.
[106] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).
[107] 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. Cape Town, South Africa, 2013.*
[108] 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.
[109] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).
[110] 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-2016)*. Association for Computational Linguistics, San Diego, California, 1011–1016.
[111] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics,

Bangkok, Thailand (online).
[112] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).
[113] Alan P. Rudell. 1993. Frequency of word usage and perceived word difficulty: Ratings of Kucera and Francis words. *Behavior Research Methods, Instruments, and Computers* 25, 4 (1993), 455–463.
[114] Irene Russo. 2021. archer at SemEval-2021 Task 1: Contextualising Lexical Complexity. In *Proceedings of the Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).
[115] 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, Sofia, Bulgaria, 103–109.
[116] Matthew Shardlow. 2013. The CW Corpus: A New Resource for Evaluating the Identification of Complex Words. In *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations*. Association for Computational Linguistics, Sofia, Bulgaria, 69–77.
[117] 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)*. European Language Resources Association, Marseille, France.
[118] Matthew Shardlow, Richard Evans, Gustavo Paetzold, and Marcos Zampieri. 2021. SemEval-2021 Task 1: Lexical Complexity Prediction. In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2021)*.
[119] Matthew Shardlow, Richard Evans, and Marcos Zampieri. 2021. Predicting Lexical Complexity in English Texts. *arXiv preprint arXiv:2102.08773* (2021).
[120] 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., Varna, Bulgaria.
[121] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).
[122] Punardeep Sikka and Vijay Mago. 2020. A Survey on Text Simplification. arXiv:2008.08612 [cs.CL]
[123] Ravi Sinha. 2012. UNT-SimpRank: Systems for Lexical Simplification Ranking. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Association for Computational Linguistics, Montréal, Canada, 493–496.
[124] Greta Smolenska. 2018. *Complex Word Identification for Swedish*. Master's thesis. Uppsala University, Sweden.
[125] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).
[126] Sanjay S.P, Anand Kumar M, and Soman K P. 2016. AmritaCEN at SemEval-2016 Task 11: Complex Word Identification using Word Embedding. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, 1022–1027.
[127] Lucia Specia, Kumar Jauhar, Sujay, and Rada Mihalcea. 2012. SemEval - 2012 Task 1: English lexical simplification. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*. Association for Computational Linguistics, Montréal, Canada, 347––355.
[128] L. Srinivasan and Chinnu Nalini. 2019. An improved framework for authorship identification in online messages. *Cluster Computing* 22, 1 (2019), 12101–12110.
[129] 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 2019)*. INCOMA Ltd., Varna, Bulgaria, 1141–1150.
[130] Regina Stodden and Gayatri Venugopal. 2021. RS_GV at SemEval-2021 Task 1: Sense Relative Lexical Complexity Prediction. In *Proceedings of the Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).
[131] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: Enhanced Representation through Knowledge Integration. arXiv:1904.09223 [cs.CL]
[132] 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 Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), Portorož, Slovenia, 230–236.
[133] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).

[134] 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.

[135] 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.

[136] Giuseppe Vettigli and Antonio Sorgente. 2021. CompNA at SemEval-2021 Task 1: Prediction of lexical complexity analyzing heterogeneous features. In *Proceedings of the Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).

[137] Katja Voskoboinik. 2021. katildakat at SemEval-2021 Task 1: Lexical Complexity Prediction of Single Words and Multi-Word Expressions in English. In *Proceedings of the Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).

[138] Sanja Štajner and Maja Popović. 2016. Can Text Simplification Help Machine Translation? *Baltic Journal of Modern Computing* 4, 2 (2016), 230–242.

[139] Robert A. Wagner and Michael J. Fischer. 1974. The String-to-String Correction Problem. 21, 1 (1974).

[140] Maolin Wang, Shervin Malmasi, and Mingxuan Huang. 2015. The Jinan Chinese Learner Corpus. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, Denver, Colorado, 118–123.

[141] 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, New Orleans, United States.

[142] 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, NY, USA.

[143] 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)*. Association for Computational Linguistics, Hong Kong, 6383–6389.

[144] 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* (Wachington, DC). 354–359.

[145] 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-2016)*. Association for Computational Linguistics, San Diego, California, 953–957.

[146] 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:2106.06090 [cs.CL]

[147] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).

[148] 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.

[149] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).

[150] 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, Santa Fe, New Mexico, USA.

[151] Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Luci 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, New Orleans, United States.

[152] 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 2017*. INCOMA Ltd., Varna, Bulgaria, 813–822.

[153] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).

[154] George-Eduard Zaharia, Dumitru-Clementin Cercel, and Mihai Dascalu. 2020. Cross-Lingual Transfer Learning for Complex Word Identification. arXiv:2010.01108 [cs.CL]

[155] 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 Fifteenth Workshop on Semantic Evaluation*. Association for Computational Linguistics, Bangkok, Thailand (online).

[156] 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.

[157] 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 2017)*. Asian Federation of Natural Language Processing, Taipei, Taiwan, 59–63.

[158] 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-2016)*. Association for Computational Linguistics, San Diego, California, 1001–1005.

[159] Qing Zeng, Eunjung Kim, Jon Crowell, and Tony Tse. 2005. A Text Corpora-Based Estimation of the Familiarity of Health Terminology. Springer-Verlag, Berlin, Heidelberg.