Master Thesis

An investigation of complex word identification (CWI) systems for English.

Adam Tucker

a thesis submitted in partial fulfilment of the requirements for the degree of

MA Linguistics
(Text Mining)

Vrije Universiteit Amsterdam

Computational Lexicology and Terminology Lab
Department of Language and Communication
Faculty of Humanities

Supervised by: Dr. Luís Morgado da Costa and Dr. Hennie van der Vliet
2nd reader: Dr. Ilia Markov

Submitted: 30 June, 2023

Abstract

Complex Word Identification (CWI) aims to identify difficult words for a reader, enabling better reading comprehension. CWI has many applications for different demographics, including language learners, people with learning difficulties, or readers with low literacy levels. This thesis explores to what extent it is possible to build and reimplement a complex word identification system based on the best-performing system from the shared task competition at CWI-2018. Models were built based on the systems that were developed at CWI-2018, and learner corpora and contextual features were added to try and improve performance. An attempt was made to recreate the winning CAMB system (Gooding and Kochmar, 2018) using publicly accessible resources. This attempt returned a lower performance than the original system but achieved an F-score of 0.79 when tested on all data compared to an average score of .84 across all data. The data used in these experiments was the *CWIG3G2* data set from (Yimam et al., 2017a), as in the original competition. The additional learner and contextual features that were added were not found to offer improvement to the performance of the original model. The original data was divided into non-native and native annotators, and all systems performed worse on the divided data.

Keywords : Complex Word Identification (CWI), NLP, Contextualized embeddings, Learner corpora, Native speakers, Non-native speakers.

Declaration of Authorship

I, Adam John Tucker, declare that this thesis, titled *An investigation of complex word identification (CWI) systems for English.* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a degree degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: 13 August 2023

Signed:

Acknowledgments

I am grateful to my family, friends, and supervisors, Dr Luís Morgado da Costa and Dr Hennie van der Vliet, for their support during the writing of this thesis. Their encouragement and faith in my abilities kept me motivated. Additionally, I am thankful to my employer, the Amsterdam School of International Business, for providing me with the necessary time and resources to pursue this academic venture alongside my job teaching English. Finally, I appreciate the assistance of my fellow students and teaching staff, who have aided me throughout the process.

List of Figures

Contents

Chapter 1

Introduction

1.1 Background

In Natural Language Processing (NLP), the task of Complex Word Identification (CWI) was originally the first stage in the task of Lexical Simplification (LS). The first LS studies did not prioritise identifying complex words as a first step but proceeded directly to word simplification. However, this method was found to be computationally intensive, as simplifying all words is unnecessary. Identifying the specific word causing difficulty for the reader was a more efficient approach. Therefore, determining which words were complex became necessary, and the task of identifying complex words became a stand-alone NLP task.

CWI machine learning (ML) models have been developed to recognise complex words that may pose challenges for children, second language learners, individuals with reading difficulties like dyslexia or aphasia, or those with limited literacy skills. For example, Carroll et al. (1999) researched simplifying all content words to aid aphasia patients with language impairments in comprehending English newspaper articles. More recently, CWI has been used for the development of authorship identification (North et al., 2023). This involves measuring vocabulary richness in a text, which can be used as a linguistic fingerprint and extract a unique writing style. For instance, research by Abdallah et al. (2013) used complex words as a feature for an email authorship identification. Other examples of CWI include those developed by Petersen and Ostendorf (2007) for text simplification for second language learners, Specia (2010) for Portuguese native speakers with low literacy levels, and Rello et al. (2013) for people with dyslexia. In short, with the recent growth in distance learning and educational technologies, ML models that automatically recognise complex words have an increasing number of applications.

1.2 Complex Word Identification (CWI)

The problem of CWI could be defined as the task of detecting words a specific target audience finds challenging to comprehend when reading. This is an issue because when a reader encounters an unknown word, they either cease reading, continue without understanding or place an incorrect meaning to the sentence. Some readers may consult a dictionary, but this action breaks the concentration, which requires further cognitive effort to reengage. Many factors contribute to the complexity of understanding specific words, depending on an individual's language background, personal experiences, and various linguistic characteristics. The challenge posed by a particular word can be a highly individual concept that relies on a reader's native language, skill level, and reading background. The perception of complexity varies significantly even among native speakers, influenced by factors such as age and education level. Furthermore, the difficulty of understanding a particular word can arise from many other factors. In linguistics, the definition of a complex word, in terms of morphology, is that it consists of two or more morphemes. A morpheme is a unit of language which can not be subdivided. A complex word includes a root and one or more affixes, such as the word *slower - er* or more than one root word in a compound, such as *blackbird*. However, an English word that contains only a root could also be complex. For instance, *script* as the root of the word *manuscript* could have many different meanings, complex or otherwise, depending on the context and semantic meaning. This leads to the second main area that linguists refer to when describing complexity. This is whether a word is semantically transparent. A word is semantically transparent if its meaning can be derived from the sum of its morphological parts. For instance, the word *un - happi - ness* as compared to *depart - ment*. In the first example, the meaning can be decoded from the morphemes. However, in the second example, the meaning of "depart" is not helpful in understanding "department". Furthermore, if the word consists of just the root and can not be broken down into morphological parts, then this could possibly add another level of difficulty for a reader.

Another factor that can lead to semantic ambiguity is the ever-changing nature of lexis over time. Neologisms and archaic words are just two such factors that can cause this. An archaic word means that it may have fallen out of common usage over time. Another factor could be that the word is borrowed from another language or refers to a concept that is not typical within the reader's culture, which may pose challenges for comprehension. Additionally, the word may be uncommon, with limited exposure for most people, thus making it harder to grasp. In some cases, the word may refer to a highly specialized concept, requiring specific knowledge or expertise to understand its meaning fully. Finally, there might be instances where a common word takes on an uncommon meaning within a given context, leading to confusion or misinterpretation for readers who are not familiar with that specific usage. Thus CWI is an important NLP task that has been researched for the last decade.

To sum up, CWI has become a popular task in its own right and not only as the primary stage in the LS pipeline. The first researcher to consider CWI as a distinct task from LS was Shardlow (2013). Two specific CWI shared task competitions have been held, one at SemEval 2016 (Paetzold and Specia, 2016b) and the other at 2018 (Yimam et al., 2018). At SemEval 2021 (Shardlow et al., 2021b), the task was renamed Lexical Complexity Prediction (LCP), as the data used for this task defined complexity on a Likert scale of 1-5 as opposed to a binary value. A definition of complexity and types of complexity in terms of CWI is expanded on in chapter 2. The Related work section will also include an analysis of the major studies and shared tasks in this area.

This thesis aims to contribute to the task of CWI by building a system inspired by the winning CAMB system (Gooding and Kochmar, 2018) from the CWI Shared Task at SemEval 2018 (Yimam et al., 2018) and investigate additional features. The CAMB system code is available on Github[1], but it does not have all the resources available to run. Lastly, this thesis will investigate the effect of additional features on the performance of the model by becoming a virtual participant at CWI-2018.

[1] https://github.com/siangooding/cwi_2018/tree/master

1.3 Research Question

The main research question is thus:

To what extent is it possible to build a system for CWI based on the best-performing CAMB system from CWI-2018, using open, publicly available resources?

The following sub-questions will also be addressed:

1) Would contextual features and additional learner corpus word frequency information help in the task of complex word identification?

2) To what extent do the complex word annotations by non-native and native speakers impact the performance of the CWI models? Is this difference measurable?

In order to answer the research questions, the research is threefold. Firstly, this thesis will create baseline systems, for binary and probabilistic classification, based on the features described in Yimam et al. (2017a). Then try to partially re-implement the best-performing CAMB system (Gooding and Kochmar, 2018) with all publicly available resources. As a result, investigating features and systems that may be useful for further research into this topic. These systems will aim to shadow participation in both the binary and probabilistic tracks for English, as the CAMB system did in the original 2018 competition. Lastly, the research will attempt to improve the performance of the system by exploring a few new features. Two areas that will be examined for features will be contextualised embeddings and learner corpus data; this is outlined in greater detail in the Method 3 section. Finally, this thesis plans to investigate a separate track that splits the non-native and native annotated data and investigates to what extent the annotations provided by these two groups have an impact on the performance of CWI models. In CWI 2018, the data was annotated by ten native and ten non-native speakers. The competition used a combined score to determine if or how complex the word was, giving a binary and probabilistic annotation from the combined non-native and native annotations. More information on the data set used for this thesis is in 2.5.2 Description of 2018 Data. In addition to participating in the task as it was originally designed, this research will run the aforementioned separate track that makes models with features aimed at improving CWI solely for non-native speakers. The model's features will be adjusted to investigate which features can improve performance for the non-native data.

The relevance of this extra track is to improve the process of CWI in the educational domain by focusing on the understanding of features that improve complexity detection for the non-native demographic. Specifically, to examine the task from the perspective of a non-native English language learner and develop a system that identifies non-native complexity identification improvement. Ideally, a CWI system useful in education would incorporate complexity with CEFR (Common European Framework of Reference for Languages). The Common European Framework of Reference (CEFR) is an internationally recognised standard for describing language proficiency based on six reference levels – A1, A2, B1, B2, C1 and C2. It aims to "provide a sound basis for the mutual recognition of language qualifications" (Europe, 2020). A system that could differentiate levels of complexity per CEFR level could be useful for teachers and learners of English. However, after some research, it was found that such a publicly available data set annotated for complexity by level is not available. A more detailed investigation and research into the available data sets will be covered in the Related Work section.

1.4 Outline

The following is a breakdown and brief overview of what will be included in the remaining chapters.

Chapter 2 Related Work presents a comprehensive review of relevant literature, theories, and studies related to CWI and defines the main concepts drawing upon existing theories, corpora and methods used.

Chapter 3 Method describes the overall design and approach adopted for the study. This includes details of the data collection and other resources used for word features. This chapter also describes machine learning methods used for classification and the description of text processing.

Chapter 4 Results presents the performance of the models described in the previous chapter and summarises the main findings to address the research questions. Finally, the section includes some information about the error analysis.

Chapter 5 Discussion analyses the results, interprets the findings, and discusses their implications and significance with respect to the research questions. Additionally, it compares the obtained results for the different systems produced in this investigation with each other and compares the results with previous CWI studies. Finally, it highlights any consistencies or disparities.

Chapter 6 Conclusion will summarise the key insights from the experiments, draw conclusions based on the results and discussion, and mention limitations and possible future directions for CWI research.

Chapter 2

Related Work

The purpose of this chapter is to give an explanation and definition of the frequently used terms in the field of CWI. It also provides information on previous research, collaborative efforts, and annotated data sets that can be used as resources for future work. This chapter will give an overview of the three prior shared tasks, examining the data they employed and analyzing the architecture of the top three performing systems at each event. Finally, it covers further CWI approaches and classification techniques.

2.1 Definition of Complexity

According to Pallotti (2015), the complexity of a word can be divided into absolute complexity or relative complexity. Absolute complexity, also known as objective complexity, is determined by the linguistic properties of a word. These properties include morphology, syntax, semantics, and phonology. Examples of linguistic characteristics that contribute to absolute complexity include having many morphemes, containing derivational or inflectional affixes, having multiple meanings, or containing multiple vowels or diphthongs. Furthermore, it is possible for words to be considered complex in one situation but not in another. This is because some words have multiple meanings. Polysemous words can often have high absolute complexity because the context often dictates the semantic meaning. An illustration of polysemy can be seen in the word 'sound'. This term has an extensive array of definitions. Specifically, it has 19 noun meanings, 12 adjective meanings, 12 verb meanings, 4 meanings in verb phrases, and 2 adverb meanings.

Relative complexity, also known as agent-related complexity, refers to the complexity of language that is influenced by an individual's experience and psycholinguistic factors(Pallotti, 2015). For example, words that refer to a specific art, culture, or historical group can be difficult for a second language learner, particularly if there are no cognates or similar cultural contexts available in their native language. A key problem with CWI is this idiosyncratic notion of complexity, which can be highly subjective. For example, for the purposes of English language education, the definition of complexity is expected to depend on the learner's level and educational background. A CWI task that addressed the level of complexity from the perspective of a language learner's ability would be ideal to address this notion of relative complexity in this particular example. Such a task would ideally have annotated data from native speakers of various languages that would allow for comparison. It would then be possible to determine if there were English words that were more complex for some learners than others. For example, Romance languages such as Spanish and Italian have roots in Latin, the same as English. Therefore, many complex words in English share similarities with their Spanish and Italian counterparts, making it easier for these learners to recognize and understand their meanings. For example, words like "communication" (comunicación/comunicazione), "information" (información/informazione). This could lower the relative complexity of a word if it was compared to a native speaker of a language such as Mandarin Chinese, which does not share these similarities. There is a more detailed explanation of specific psycholinguistic features used in this thesis later in this chapter2.1.1.

More recently, North et al. (2023) state there are four types of complexity for CWI: comparative, binary, continuous and personalised. Comparative complexity prediction is a sub-task of lexical simplification (LS) that uses a value to differentiate between target words based on their level of complexity. This type of word complexity was used at the SemEval-2012(Specia et al., 2012) LS task, which in turn came from the SemEval-2007 (McCarthy and Navigli, 2007) shared task on Lexical Substitution. Between 2012 and 2018, binary complexity prediction was the main area of focus in complexity prediction research. This involves assigning a binary complexity value of 1 or 0 to a target word, indicating whether it is complex or non-complex. Research has shown that binary CWI can have low inter-annotator agreement due to the subjectivity of lexical complexity, which, as mentioned, depends on an individual's experiences and prior knowledge (Maddela and Xu, 2018).

The introduction of Lexical Complexity Prediction (LCP) moves away from binary classification to a rating system. This aims to handle words that have an unclear level of complexity, as well as words that fall on the decision boundary. LCP measures complexity based on a range of difficulty levels and uses this information to make predictions. It assigns varying levels of complexity to target words, such as the Likert scale used at SemEval-2021. Continuous complexity, also known as LCP (Lexical Complexity Prediction), involves measuring the degree of difficulty associated with a particular word and predicting its complexity level accordingly. This is accomplished by assigning a complexity label to each target word, ranging from very easy to very hard, based on specific thresholds. The scale on which the words were rated was for this specific task and data set used at SemEval-2021, and the thresholds are defined as follows: very easy (0), easy (0.25), neutral (0.5), difficult (0.75), or very difficult (1). These scales could have been with 10 points or 100 points, but this was the way the data was annotated in this task.

Researchers studying complexity prediction are exploring ways to personalize lexical simplification (Lee and Yeung, 2018). Prior systems have not taken into account the variations in vocabulary knowledge among users, resulting in a "one size fits all" approach that fails to accurately model varying perceptions of lexical complexity. To address this issue, personalized CWI systems have been introduced, which cater to individual users or specific target demographics. These systems use demographic features such as language proficiency, native language, race, job, age, ethnicity, or education to make predictions on an individual basis. For example, (Tack et al., 2016) predicted the lexical competence of French as a foreign language learners (FFL) by targeting learners who were native speakers of Dutch having attained the A2/B1 proficiency in French.

Lastly, there is the Personalised Data Set from (Lee and Yeung, 2018), which uses 15 learners of English who were native Japanese speakers. This data consists of 12,000 words labelled with varying degrees of complexity, using a 5-point Likert scale. The most recent data set is the CompLex data set (Shardlow et al., 2020) contains 10,800 words and MWEs labelled with 5-point Likert with the words given in context taken from the Bible, biomedical articles, and Europarl. This set was annotated by people from US, UK, and Australia.

2.1.1 Psycholinguistic Features Description

The Medical Research Council (MRC) Psycholinguistic Database [1] of English words (Wilson, 1988; Coltheart, 1981) has data that is freely available and in accessible format for NLP. Psycholinguistic attributes are characteristics of words that affect how they are processed and understood in the human brain. Some frequently encountered psycholinguistic attributes of words are frequency, imageability, phonological properties, semantic relatedness and morphological complexity. The frequency of a word refers to how often it is used in a language. Generally, words that are used more often are processed more accurately and quickly than less common words. Concreteness refers to how well a word represents a tangible object or idea. Words that represent physical objects, such as "apple" or "dog," are generally processed more accurately and quickly than abstract words, such as "freedom" or "justice." Imageability is the ability of a word to bring to mind mental images or visual associations. Words that are highly imageable, like "mountain" and "rainbow," are usually processed more quickly and accurately than words with low imageability, such as "justice" and "philosophy." Phonological properties relate to how words sound, such as their syllable structure, stress pattern, and phoneme composition. Words that are easy to pronounce and have a consistent sound structure are usually processed faster and more accurately than words with irregular phonological properties. Semantic relatedness refers to how closely a word's meaning is connected to other words in a language. When words are semantically related, they are usually processed faster and with greater accuracy compared to words that have no relation to each other. Morphological complexity refers to the number of morphemes in a word. Morphemes are the smallest units of meaning in a language. Words with more morphological complexity, such as "unhappiness," are usually processed slower and less accurately than simpler words, such as "happy."

2.2 Evaluation methods used in CWI systems

This section will discuss the metrics used to evaluate the performance of systems used in the three international competitions at SemEval-2016, CWI–2018 at BEA and SemEval-2021. These metrics include accuracy, precision, recall, F1-score, G-score, mean absolute error, mean squared error, Pearson's correlation, and Spearman's rank.

F1-Score.F1-score is the harmonic average of the accuracy and recall scores. It penalizes those systems that demonstrate either low precision and recall or a high imbalance between the two. For each class, the F1-scores are computed and then used to determine the macro and weighted F1-scores for all the systems. The macro F1-score is the average of all per-class F1-scores, while the weighted F1-score considers the number of actual occurrences of each class within the dataset to determine the average of all per-class F1-scores. The F1 score is calculated using the formula below:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.1}$$

[1] `https://websites.psychology.uwa.edu.au/school/mrcdatabase/uwa_mrc.htm`

G-score. G-score or Geometric mean unlike F1-score, takes into account accuracy and recall rather than precision and recall (the square root of the product of precision and recall).

$$G = \sqrt{\frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FP})\cdot(\mathrm{TP}+\mathrm{FN})}} \tag{2.2}$$

Mean absolute error (MAE) This is used for systems that are required to predict continuous instead of binary complexity values are commonly evaluated using mean absolute error, mean squared error, Pearson Correlation, and Spearman'Rank. The Mean Absolute Error (MAE) is calculated as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \tag{2.3}$$

Pearson's Correlation In LCP-2021, the main method of evaluation was Pearson's Correlation (R). This measures the linear connection between two variables, resulting in a value between 1 and 1. A value closer to 1 indicates a strong positive correlation, while a value closer to -1 indicates a strong negative correlation. The equation used for calculating Pearson's correlation is as follows.

$$R_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X\sigma_Y} \tag{2.4}$$

Spearman's Rank Spearman's Rank (ρ) considers the non-linear relationship between two variables, making it more robust in handling outliers compared to Pearson's Correlation. The value it returns ranges between -1 and 1, which indicates the corresponding correlations,strong negative (-1) and strong positive. It is calculated using the following equation.

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{2.5}$$

In the equation *di* is the difference between the two ranks of each observation, and n is the total number of observations.

2.3 Previous tasks on Complex Word Identification

This section will examine the previous SemEval (International Workshop on Semantic Evaluation) CWI shared tasks from 2016 (Paetzold and Specia, 2016b), CWI-2018 (Yimam et al., 2018), and the Lexical Complexity Prediction task in 2021 (Shardlow et al., 2021b). It was found that This section will then go on to describe each of these shared task competitions, the best-performing systems that were submitted, as well as any other systems and data that could be applicable.

Prior to the 2016 shared task, NLP research was not focused on CWI as a standalone task. At SemEval-2012, CWI was the first major component for the task of lexical simplification, but it was not a dedicated task on its own and did not have a large annotated data set until 2016 (Specia et al., 2012). The LS-2012 data set contains 201 complex words, each with multiple options for simplification and each complex word is shown in 10 different contexts. Native English speakers annotated the data set by providing suggestions for simplification, and four L2 learners ranked the proposed simplifications based on their level of complexity. Next, in 2013, there was the CW Corpus from Shardlow. This corpus contained 731 complex words and their equivalent simplification. Complex words were taken from Wikipedia edit history, editor comments, and a series of simplification checks. This data set had 500 lexical simplification examples and used 50 annotators from the US. Complex words were taken from Wikipedia and provided in context at the sentence level.

Another useful resource for CWI and LS is the Word Complexity Lexicon [2] as described by (Maddela and Xu, 2018). They created a data set of human annotations for the 15,000 most frequent English words in Google 1T Ngram Corpus. These data sets used a 6-point Likert scale with the categories: very easy, very hard, moderately easy, moderately hard, easy and hard and were annotated by 11 non-native yet fluent English speakers. The scores consist of English words and their complexity scores obtained by averaging over the human ratings. The data is readily available and is open access.

A study titled "Lexical Simplification for Non-Native English Speakers," conducted by (Paetzold, 2016), found that non-native English speakers tend to perceive certain words as complex based on their level of proficiency in the language. The study also discovered that this perception of complexity decreases with age. Furthermore, the research emphasized that although such words may not be commonly found in corpora and may be less ambiguous, their level of complexity can vary based on the context in which they are used. One of the methods used to distinguish complex words was detecting their presence in a lexicon, using the word lists Ogden's Basic English (Ogden and Halász, 1935), the Simple Wikipedia (Kauchak, 2013), and the SubIMDB corpus (Paetzold and Specia, 2015), which consists of 4,351 subtitles of movies and series for children.

2.4 SemEval-2016 Shared Task on CWI

2.4.1 Task

The first major shared task for CWI was held at the International Workshop on Semantic Evaluation (SemEval-2016) and was Task 11: Complex Word Identification [3]. 42 systems were submitted from 21 distinct teams, and nine baselines were provided. The data used for this task is free and available for potential further research. The objective of the task was to predict the complexity of a word for someone who is not a native speaker, utilising the annotations of non-native speakers. Table 2.2 shows the 10 best-performing systems ranked in order of G-score. In 2016, the organisers evaluated the systems using G-score (Equation (2.2)). A full list of the 42 systems and baseline system scores are shown in Table 2.1. The motivation for using G-Score for evaluation as opposed to F1 was that the researchers felt the F1 evaluation did not accurately capture the effectiveness of a solution for the task in terms of CWI for LS (Paetzold and Specia, 2016b).

[2] https://github.com/mounicam/lexical_simplification/tree/master

[3] SemEval-2016 Complex Word Identification https://alt.qcri.org/semeval2016/task11/

2.4.2 Description of 2016 Data

A group of 400 non-native English speaking participants were enlisted to annotate the data. The majority of the participants were either university students or staff members. The participants provided information about their native language, age, education level, and English proficiency level based on the Common European Framework of Reference for Languages (CEFR). They were required to determine if they could comprehend the meaning of each content word (nouns, verbs, adjectives, and adverbs) in a group of sentences. The sentences were judged individually, and volunteers were asked to highlight all the words they did not understand, even if they comprehended the sentence as a whole. The 200 sentences were separated into 20 groups of 10 sentences, and 20 volunteers annotated each group. The remaining 9,000 sentences were divided into 300 groups of 30 sentences, and each group was annotated by a single person. The annotators spoke 45 different languages, with the most common being Portuguese (15.3%), Chinese (13%), and Spanish (11.3%). The annotators' age range was between 18 and 66 years old, with an average age of 28.2. Of the volunteers, 63.7% were postgraduate students, 32.3% were undergraduate students, and 4% were in high school. 36.8% of the annotators claimed to have advanced (C2) English proficiency skills, 37.7% claimed to have pre-advanced (C1) skills, 16.6% claimed to have upper-intermediate (B2) skills, 6.4% claimed to have intermediate (B1) skills, 2% claimed to have pre-intermediate (A2) skills, and 0.5% claimed to have elementary (A1) skills. (Paetzold and Specia, 2016b)

2.4.3 Analysis of best-performing systems in 2016

The results of the shared task showed that ensemble methods trained on morphological, lexical, and semantic features outperformed any other ML technique, including systems that used neural approaches. Also, the more straightforward features based on word frequency and word presence in certain lexicons worked best. The research found that "word frequencies remain the most reliable predictor of word complexity." Paetzold and Specia (2016b) Furthermore, the findings demonstrate that Decision Trees and Ensemble methods are effective for the task, but word frequencies are still the most dependable predictor of word complexity. The top three performing systems are analysed below. Table 2.2 show the best-performing system's scores SV000G, TALN, UWB. Overall, it can be said that systems in 2016 did not perform well in terms of F1 score.

The system with the highest F-1 score was the PLUJAGH team with their second SEWDFF system (not shown in Table 2.2 that scored 0.922 Accuracy 0.289 Precision, 0.453 Recall, 0.353 F-score and had a G-Score of 0.608. When evaluating a classifier's performance, using the geometric mean instead of the harmonic mean will favour precision and recall values that are closer together. This is because the harmonic mean F1 score tends to be more pessimistic than the geometric mean.

PLUJAGH-SEWDFF (Wróbel, 2016) was a simple rule-based method that used information regarding whether the target word was in a prepared vocabulary list. They presented two Threshold-Based solutions to CWI that outperformed the rest. Threshold-based approaches aim to find a specific threshold (t) for a given metric of simplicity (M) that can accurately categorise a word (w) as complex or simple. If the metric value for the word M(w) ¡ t, the word is classified as a simple (or complex) word. Implementing threshold-based approaches is both intuitive and simple and in

G	F	Team	System	Accuracy	Precision	Recall	F-score	G-score
22	1	PLUJAGH	SEWDFF	0.922	0.289	0.453	0.353	0.608
16	2	LTG	System2	0.889	0.22	0.541	0.312	0.672
36	3	LTG	System1	0.933	0.3	0.321	0.31	0.478
26	4	MAZA	B	0.912	0.243	0.42	0.308	0.575
6	5	HMC	DecisionTree25	0.846	0.189	0.698	0.298	0.765
-	-	Baseline	(HV) No Baselines	0.88	0.204	0.539	0.296	0.668
9	6	TALN	RandomForest_SIM	0.847	0.186	0.673	0.292	0.75
5	7	HMC	RegressionTree05	0.838	0.182	0.705	0.29	0.766
8	8	MACSAAR	RFC	0.825	0.168	0.694	0.27	0.754
3	9	TALN	RandomForest_WEI	0.812	0.164	0.736	0.268	0.772
4	10	UWB	All	0.803	0.157	0.734	0.258	0.767
4	11	PLUJAGH	SEWDF	0.795	0.152	0.741	0.252	0.767
-	-	Baseline	(HV) All Systems	0.791	0.151	0.748	0.251	0.769
7	12	JUNLP	RandomForest	0.795	0.151	0.73	0.25	0.761
1	13	SV000gg	Soft	0.779	0.147	0.769	0.246	0.774
10	14	MACSAAR	NNC	0.804	0.146	0.66	0.24	0.725
4	15	JUNLP	NaiveBayes	0.767	0.139	0.767	0.236	0.767
2	16	SV000gg	Hard	0.761	0.138	0.787	0.235	0.773
32	17	USAAR	entropy	0.869	0.148	0.376	0.212	0.525
17	18	MAZA	A	0.773	0.115	0.578	0.192	0.661
31	19	BHASHA	DECISIONTREE	0.836	0.118	0.387	0.181	0.529
34	20	BHASHA	SVM	0.844	0.119	0.363	0.179	0.508
11	21	Pomona	NormalBag	0.604	0.095	0.872	0.171	0.714
12	22	Melbourne	runw15	0.586	0.091	0.87	0.165	0.701
13	23	UWB	Agg	0.569	0.089	0.885	0.161	0.693
14	24	Pomona	GoogleBag	0.568	0.088	0.881	0.16	0.691
28	24	GARUDA	SVMPP	0.796	0.099	0.415	0.16	0.546
15	25	IIIT	NCC	0.546	0.084	0.88	0.154	0.674
16	25	Baseline	(TB) Wikipedia	0.536	0.084	0.901	0.154	0.672
24	26	ClacEDLK	ClacEDLK-RF_0.5	0.751	0.09	0.475	0.152	0.582
40	27	GARUDA	HSVM&DT	0.88	0.112	0.226	0.149	0.36
18	28	Baseline	(TB)SimpleWiki	0.513	0.081	0.902	0.148	0.654
19	29	Melbourne	runw3	0.513	0.08	0.895	0.147	0.652
37	29	USAAR	entroplexity	0.834	0.097	0.305	0.147	0.447
21	30	ClacEDLK	ClacEDLK-RF_0.6	0.688	0.081	0.548	0.141	0.61
20	31	Sensible	Baseline	0.591	0.078	0.713	0.14	0.646
23	32	IIIT	NCC2	0.465	0.071	0.86	0.131	0.604
25	33	Baseline	(TB) Senses	0.436	0.068	0.861	0.125	0.579
33	34	Sensible	Combined	0.737	0.072	0.39	0.122	0.51
27	35	AmritaCEN	w2vecSim	0.627	0.061	0.486	0.109	0.547
41	35	CoastalCPH	Concatenation	0.869	0.08	0.171	0.109	0.285
35	36	CoastalCPH	NeuralNet	0.693	0.063	0.398	0.108	0.506
36	37	Baseline	(TB) Length	0.332	0.057	0.852	0.107	0.478
39	38	Baseline	(LB) Ogdens	0.248	0.056	0.947	0.105	0.393
29	39	AIKU	native1	0.583	0.057	0.512	0.103	0.545
29	40	AIKU	native	0.555	0.056	0.535	0.101	0.545
30	41	AKTSKI	wsys	0.587	0.056	0.49	0.1	0.534
38	41	AmritaCEN	w2vecSimPos	0.743	0.06	0.306	0.1	0.434
30	42	AKTSKI	svmbasic	0.512	0.053	0.558	0.097	0.534
42	43	Baseline	(LB) Wikipedia	0.047	0.047	1	0.089	0.09
43	43	Baseline	All Complex	0.047	0.047	1	0.089	0.089
44	44	Baseline	(LB)SimpleWiki	0.953	0.241	0.002	0.003	0.003
45	45	Baseline	All Simple	0.953	0	0	0	0

Table 2.1: Results for all systems from Sem-Eval 2016 Paetzold and Specia (2016b)

2016, produced the best performance (Paetzold, 2016). The SEWDFF system considers a word to be complex if its frequency in Simple Wikipedia is less than 147. Features used were: term frequency and document frequency for the word and its lemma use in English Wikipedia, Simple English Wikipedia and corpora created from training and test sentences. Additionally, the system used the length of the sentence (number of words), length of the word (number of characters), the position of the word in the sentence, and GloVe word embedding (Pennington et al., 2014).

The SV000gg-Soft and SV000gg-Hard systems, as discussed in Paetzold and Specia (2016c), were the top performers in terms of G-score. These systems utilised ensemble-based models that incorporated various sub-models. The developers believed that using diverse models would enhance the CWI performance. They conducted experiments using ensemble-based models that incorporated a lexicon-based model and a threshold-based model for Support Vector Machines, Decision Trees, and Random Forests. The lexicon-based model was used to determine if a target word was complex or non-complex by searching for it in a pre-labelled dictionary of lexemes. The threshold-based model was used to separate complex and non-complex words based on specific features that were found to be defining characteristics of each word type and if the target word had a feature above a certain threshold (North et al., 2023).

TALN (RandomForest_WEI) (Ronzano et al., 2016)used Random Forests, which consisted of several Decision trees trained on multiple features. For features, they used 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) and the 2014 English Wikipedia Corpus.

The UWB system (Konkol, 2016) used the Maximum Entropy classifier for their system but stated the choice of classifier only had minimal impact compared with the choice of features. They concluded that word frequency and document frequency were the best CWI predictors. Their final system used the single feature of document frequency. With just this feature, they ended up as the third-best team (with the 4th best system). This clearly demonstrated a state-of-the-art G-score in the 2016 CWI task was possible with a very simple system.

G	F	Team	System	Accuracy	Precision	Recall	F1-score	G-score
1	13	SV000gg	Soft	0.779	0.147	0.769	0.246	0.774
2	16	SV000gg	Hard	0.761	0.138	0.787	0.235	0.773
3	9	TALN	RandomForest_WEI	0.812	0.164	0.736	0.268	0.772
4	10	UWB	All	0.803	0.157	0.734	0.258	0.767
4	11	PLUJAGH	SEWDF	0.795	0.152	0.741	0.252	0.767
4	15	JUNLP	NaiveBayes	0.767	0.139	0.767	0.236	0.767
5	7	HMC	RegressionTree05	0.838	0.182	0.705	0.290	0.766
6	5	HMC	DecisionTree25	0.846	0.189	0.698	0.298	0.765
7	12	JUNLP	RandomForest	0.795	0.151	0.730	0.250	0.761
8	8	MACSAAR	RFC	0.825	0.168	0.694	0.270	0.754
9	6	TALN	RandomForest_SIM	0.847	0.186	0.673	0.292	0.750
10	14	MACSAAR	NNC	0.804	0.146	0.660	0.240	0.725

Table 2.2: Top 10 performing systems in order of G-score from SemEval-2016 CWI task (Paetzold and Specia, 2016b).

2.5 2018 Shared Task on CWI

2.5.1 Description of 2018 task

The second CWI task, known as CWI-2018, took place during the BEA (Building Educational Applications) Workshop at NAACL-HLT'2018 (North American Chapter of the Association for Computational Linguistics: Human Language Technologies). The objective of the 2018 CWI shared task was to use annotations from both native and non-native speakers to anticipate which words pose difficulties for those who are not native speakers. This was done because of the findings from the prior shared task in 2016; the data was found to have strong biases and inconsistencies in the test set, resulting in very low F-scores across all systems (Paetzold and Specia, 2016a; Wróbel, 2016). The 2018 competition builds on the work from 2016. The data used for the task was the CWIG3G2 [4] (Three Text Genres and Two User Groups) from Yimam et al. (2017a). In comparison to CWI-2016, CWI-2018 had three significant updates. Firstly, it was multilingual, unlike its predecessor, which was only available in English. Secondly, it allowed users to input both single and multiple words as targets. Lastly, there were two sub-tasks: one that required binary classification and another that required probabilistic classification. In 2018, CWI received submissions from 12 teams for various tasks and tracks. Following the competition, 10 teams presented their system description papers at the BEA workshop.

The objective of the 2018 CWI shared task was to determine which words pose difficulties for non-native speakers, using annotations gathered from both native and non-native speakers. In order to train their systems, participants were given a training set that had been labelled with annotations indicating the complexity of words in context with binary and probabilistic scores. A month later, an unlabeled test set was provided, and the participating teams were asked to upload their predictions for evaluation. In the CWI challenge, participants were given data sets in four languages: English, German, Spanish monolingual CWI and Multilingual with a French test set. Participants were only given a French test set, without any French training. This was done in order to strengthen the cross-lingual element. Each team were offered each of the sets as a separate track, and then each track was subdivided into the already mentioned two classification sub-tasks for the binary and probabilistic annotations (Yimam et al., 2018).

2.5.2 Description of 2018 data

For the English data set, annotations were collected using Amazon Mechanical Turk (MTurk). The annotators received a paragraph-level task HIT (Human Intelligence Task) and were instructed not to choose determiners, numbers, or phrases longer than 50 characters. They were presented with 5 to 10 sentences and were requested to highlight complex words or phrases that were complex in the paragraph. Annotators were told they should assume a given target reader, such as children, language learners or people with reading impairments, when assessing if the target is complex. More than 20 annotators annotated the same text. Complex words were not highlighted in advance to reduce bias in selection. As shown in Table 2.3, the total number of annotators was 183, consisting of 134 native speakers and 49 non-native speakers. This

[4] Complex Word Identification (CWI) Shared Task 2018 `https://sites.google.com/view/cwisharedtask2018/`

non-native category does not have a further breakdown of the native language. Yimam et al. (2017a) sorted the assignments completed by both native and non-native speakers based on their inter-annotator agreement scores and selected the top 10 for each group. The annotations showed a measurable difference in annotation agreement between the two groups.

Language	Native	Non-native	Total
English	134	49	183
German	12	11	23
Spanish	10	12	22

Table 2.3: The number of annotators for different languages (Yimam et al., 2018).

A breakdown of the data can be seen in Table 2.4, the training set contains 27,296 instances, the development set has 3,325 instances, and the test set has 4,100 instances. The data includes a considerable number of Multiple Word Expressions (MWEs), while approximately 85% of the data comprises single words. It is also worth noting that the three genres of data do not contain an equal amount of instances with the NEWS training data having 14,001 total instances with WIKINEWS and WIKIPEDIA having 7,745 and 5,550 total instances respectively.

Data	Train	Dev	Test
NEWS (Word)	11,948	1,501	1,812
NEWS (MWEs)	2,053	262	282
WIKINEWS (Word)	6,779	775	1,137
WIKINEWS (MWEs)	966	94	149
WIKIPEDIA (Word)	4,832	605	749
WIKIPEDIA (MWEs)	718	88	120
Total Instances	27,296	3,325	4,100

Table 2.4: 2018 CWI data showing number of single word and Multi Word Expression (MWE) instances (Yimam et al., 2018).

The format of the CWIG3G2 data is shown in Table 2.5. Each line represents a sentence with one complex word annotation and relevant information, each separated by a TAB character. The first column gives the HIT ID and shows the paragraph given to the annotators. All sentences with the same ID belong to the same HIT. The next column gives the sentence, and the following two columns provide the start and end slice index of the target word or phrase in the next column. The next two columns show the Native and Non-native annotator count, which is always ten each for all data. The "Native" and "Non-Native" columns give a number for how many annotators decided the word or phrase was complex for each group. The "Gold Binary" column is one if any annotators deem the word complex. The "Gold Prob" column is the probability of the word calculated from the total number of annotators divided by the total number. For instance, in the first row of the table, "flexed their muscles" was labelled complex by 3 Native speakers and 2 Non-natives; therefore, the probability was calculated as (3+2/20), giving 0.25. Additionally, the sentence includes a reference number of the paragraph that was shown to the annotators. For English, a development and training set is given for the three genres, WIKIPEDIA, WIKINEWS and NEWS. In the original competition, a test set was later supplied for each genre. Further analysis of this data is done in the Method 3.1 section, as this data was used in all experiments.

HIT	Sentence	Start	End	Target	Native	Non-native	Native	Non-native	Gold Binary	Gold Prob
3P7RGT LO6EE0 7HLUVD KKHS6O 7CCKA5	Both China and the Philippines flexed their muscles on Wednesday.	31	51	flexed their muscles	10	10	3	2	1	0.25
3P7RGT LO6EE0 7HLUVD KKHS6O 7CCKA5	Both China and the Philippines flexed their muscles on Wednesday.	31	37	flexed	10	10	2	6	1	0.4
3P7RGT LO6EE0 7HLUVD KKHS6O 7CCKA5	Both China and the Philippines flexed their muscles on Wednesday.	44	51	muscles	10	10	0	0	0	0

Table 2.5: 2018 shared task CWI training data format (CWIG3G2)Yimam et al. (2017a)

2.5.3 Best-performing system 2018

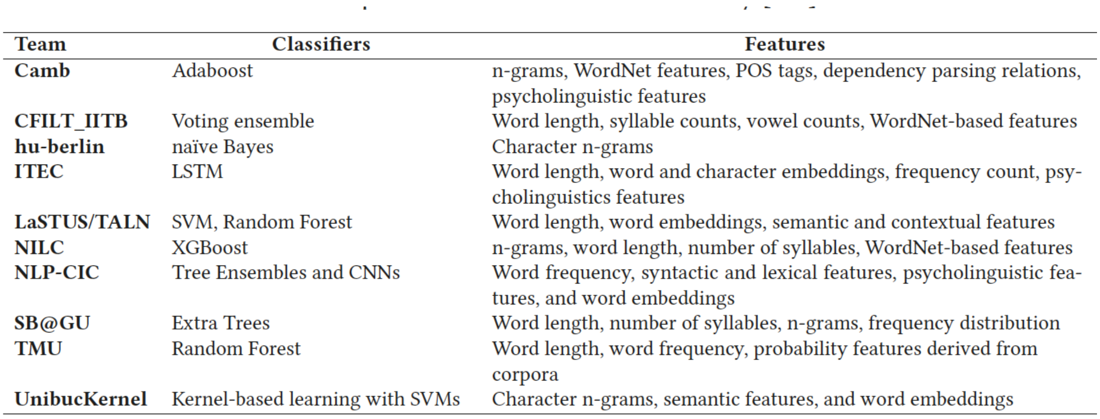
At CWI-2018, the three best-performing systems were CAMB (Gooding and Kochmar, 2018), NILC (Hartmann and Dos Santos, 2018) and ITEC (De Hertog and Tack, 2018). Figure 2.1 below shows the 10 teams that submitted description papers and the type of features and classifiers used. Table 2.6 shows all systems that were submitted for the binary classification mono-lingual English task ranked by F1 score. Table 2.7 shows the teams ranked in MAE order for the probabilistic classification task. The best-performing systems were examined to try and establish common features used. As can be seen in Table 2.6, the CAMB system outperformed all other systems across all three genres. The CAMB system uses a total of 27 features Gooding and Kochmar (2018)k: word length, number of syllables, WordNet features such as the number of synsets), word n-gram and POS tags, and dependency parse relations, the number of words grammatically related to the target word, CEFR levels, Google N-gram word frequencies and psycholinguistic features from the MRC Database Wilson (1988); word familiarity rating, number of phonemes, Thorndike-Lorge written frequency, imageability rating, concreteness rating, number of categories, samples, and written frequencies, and age of acquisition. A detailed breakdown of the features used in the CAMB system is in the Method 3.3. There are two notebooks on GitHub [5], but currently, these are no links to the external resources that were used for feature extraction.

NILC (Hartmann and Dos Santos, 2018) used Our three groups of features. Lexical: includes word length, number of syllables, number of senses, hypernyms and hyponyms in WordNet.N-gram: includes log probabilities of an n-gram containing target words in two language models trained on Book Corpus and One Billion Word data sets using SRILM (Stolcke, 2002). Lastly, they used the psycholinguistic features of familiarity, age of acquisition, concreteness and imageability. They used the XGBoost classifier for this system. This team did develop two other systems that used a shallow neural

[5] CAMB System Notebooks `https://github.com/siangooding/cwi_2018/tree/master`

network method using only word embeddings and a Long Short-Term Memory (LSTM) language model, but they did not perform as well as the feature engineered system.

ITEC (De Hertog and Tack, 2018) was a system that uses deep learning architecture with information on five aspects: distributional information of the target word itself, morphological structure, psychological measures, corpus counts and topical information. The encoding for each target word is done using its word embedding. In the case of word groups, the embeddings are concatenated. The assumption is that words with similar distributional patterns have a similar level of complexity. To reduce the dimensionality of the representation, an LSTM layer with 64 dimensions is used.



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

Figure 2.1: Systems submitted to CWI-2018 for the English binary classification task (Single word track) in alphabetical order, as summarised by Yimam et al. (2018)

2.6 SemEval-2021 Task 1: Lexical Complexity Prediction

2.6.1 Task

Task 1 at SemEval-2021 (The 15th International Workshop on Semantic Evaluation) was Lexical Complexity Prediction (LCP), also referred to as LCP-2021 [6]. The LCP shared task 2021 provided participants with a new annotated English data set with a Likert scale annotation. This data set aims to address some idiosyncratic and subjective notions of complexity by introducing a Likert scale and giving words a rating of complexity. The competition had a total of 58 teams, and the competition had two sub-tasks. Task 1 was broken down into two sub tasks. For the first sub-task, teams were asked to predict the complexity values for individual words. The second sub-task required participants to predict complexity values for the entire data set, which included MWEs (Shardlow et al., 2021b). The results of the 10 best-performing systems can be seen in Table 2.8.

[6] https://sites.google.com/view/lcpsharedtask2021

NEWS			**WIKINEWS**			**WIKIPEDIA**		
System	**F-1**	**Rank**	**System**	**F-1**	**Rank**	**System**	**F-1**	**Rank**
Camb	0.8736	1	Camb	0.84	1	Camb	0.8115	1
Camb	0.8714	2	Camb	0.8378	2	NILC	0.7965	2
Camb	0.8661	3	Camb	0.8364	4	UnibucKernel	0.7919	3
ITEC	0.8643	4	Camb	0.8378	3	NILC	0.7918	4
ITEC	0.8643	4	NLP-CIC	0.8308	5	Camb	0.7869	5
TMU	0.8632	6	NLP-CIC	0.8279	6	Camb	0.7862	6
ITEC	0.8631	7	NILC	0.8277	7	SB@GU	0.7832	7
NILC	0.8636	5	NILC	0.8270	8	ITEC	0.7815	8
NILC	0.8606	9	NLP-CIC	0.8236	9	SB@GU	0.7812	9
Camb	0.8622	8	CFILT IITB	0.8161	10	UnibucKernel	0.7804	10
NLP-CIC	0.8551	10	CFILT IITB	0.8161	10	Camb	0.7799	11
NLP-CIC	0.8503	12	CFILT IITB	0.8152	11	CFILT IITB	0.7757	12
NLP-CIC	0.8508	11	CFILT IITB	0.8131	12	CFILT IITB	0.7756	13
NILC	0.8467	15	UnibucKernel	0.8127	13	CFILT IITB	0.7747	14
CFILT IITB	0.8478	13	ITEC	0.8110	14	NLP-CIC	0.7722	16
CFILT IITB	0.8478	13	SB@GU	0.8031	15	NLP-CIC	0.7721	17
CFILT IITB	0.8467	14	NILC	0.7961	17	NLP-CIC	0.7723	15
SB@GU	0.8325	17	NILC	0.7977	16	NLP-CIC	0.7723	15
SB@GU	0.8329	16	CFILT IITB	0.7855	20	SB@GU	0.7634	18
Gillin Inc.	0.8243	19	TMU	0.7873	19	TMU	0.7619	19
Gillin Inc.	0.8209	24	SB@GU	0.7878	18	NILC	0.7528	20
Gillin Inc.	0.8229	20	UnibucKernel	0.7638	23	UnibucKernel	0.7422	24
Gillin Inc.	0.8221	21	hu-berlin	0.7656	22	hu-berlin	0.7445	22
hu-berlin	0.8263	18	SB@GU	0.7691	21	SB@GU	0.7454	21
Gillin Inc.	0.8216	22	LaSTUS/TALN	0.7491	25	UnibucKernel	0.7435	23
UnibucKernel	0.8178	26	LaSTUS/TALN	0.7491	25	LaSTUS/TALN	0.7402	25
UnibucKernel	0.8178	26	SB@GU	0.7569	24	LaSTUS/TALN	0.7402	25
CFILT IITB	0.8210	23	hu-berlin	0.7471	26	NILC	0.7360	26
CFILT IITB	0.8210	23	Gillin Inc.	0.7319	28	hu-berlin	0.7298	27
hu-berlin	0.8188	25	Gillin Inc.	0.7275	30	CoastalCPH	0.7206	28
UnibucKernel	0.8111	28	Gillin Inc.	0.7292	29	LaSTUS/TALN	0.6964	29
NILC	0.8173	27	Gillin Inc.	0.7180	31	Gillin Inc.	0.6604	30
LaSTUS/TALN/TALN	0.8103	29	LaSTUS/TALN	0.7339	27	Gillin Inc.	0.6580	31
LaSTUS/TALN	0.8103	29	Gillin Inc.	0.7083	32	Gillin Inc.	0.6520	32
LaSTUS/TALN	0.7892	31	UnibucKernel	0.6788	33	Gillin Inc.	0.6329	33
UnibucKernel	0.7728	33	SB@GU	0.5374	34	SB@GU	0.5699	34
SB@GU	0.7925	30	-	-	-	CoastalCPH	0.5020	35
SB@GU	0.7842	32	-	-	-	LaSTUS/TALN	0.3324	36
LaSTUS/TALN	0.7669	34	-	-	-	-	-	-
UnibucKernel	0.5158	36	-	-	-	-	-	-
SB@GU	0.5556	35	-	-	-	-	-	-
LaSTUS/TALN	0.2912	37	-	-	-	-	-	-
LaSTUS/TALN	0.1812	38	-	-	-	-	-	-
LaSTUS/TALN	0.1761	39	-	-	-	-	-	-
Baseline	0.7579	-	Baseline	0.7106	-	Baseline	0.7179	-

Table 2.6: Binary classification results for the monolingual English tracks. (Yimam et al., 2018)

News			WikiNews			Wikipedia		
Rank	MAE	Source	Rank	MAE	Source	Rank	MAE	Source
1	0.051	TMU	1	0.0674	Camb	1	0.0739	Camb
2	0.0539	ITEC	1	0.0674	Camb	2	0.0779	Camb
3	0.0558	Camb	2	0.0690	Camb	3	0.0780	Camb
4	0.056	Camb	3	0.0693	Camb	4	0.0791	Camb
5	0.0563	Camb	4	0.0704	TMU	5	0.0809	ITEC
6	0.0565	Camb	5	0.0707	ITEC	6	0.0819	NILC
7	0.0588	NILC	6	0.0733	NILC	7	0.0822	NILC
8	0.0590	NILC	7	0.0742	NILC	8	0.0844	Camb
9	0.1526	SB@GU	8	0.0820	Camb	9	0.0931	TMU
10	0.2812	Gillin Inc.	9	0.1651	SB@GU	10	0.1755	SB@GU
11	0.2872	Gillin Inc.	10	0.2890	Gillin Inc.	11	0.2461	NILC
12	0.2886	Gillin Inc.	11	0.3026	Gillin Inc.	12	0.3156	Gillin Inc.
13	0.2958	NILC	12	0.3040	Gillin Inc.	13	0.3208	Gillin Inc.
14	0.2978	NILC	13	0.3044	Gillin Inc.	14	0.3211	Gillin Inc.
15	0.3090	Gillin Inc.	14	0.3190	Gillin Inc.	15	0.3436	Gillin Inc.
16	0.3656	SB@GU	15	0.3203	NILC	16	0.3578	NILC
17	0.6652	NILC	16	0.3240	NILC	17	0.3819	NILC
-	0.1127	Baseline	-	0.1053	Baseline	-	0.1112	Baseline

Table 2.7: Probabilistic classification results for the monolingual English tracks (Yimam et al., 2018).

2.6.2 CompLex Data Description

Participants were provided with an augmented version of the CompLex data set [7] (Shardlow et al., 2020). The sentences were annotated with a 5-point Likert scale. The complex words for annotation came from three sources/domains: the Bible, Europarl, and biomedical texts. The annotation process resulted in a corpus of 9,476 sentences, each annotated by around seven annotators. Significantly, from the perspective of CWI for English language learners, they selected annotators from English-speaking countries. The data labels only nouns and multi-word expressions; the annotations were restricted to those that followed a Noun-Noun or Adjective-Noun structure and employed Stanford CoreNLP's POS tagger Manning et al. (2014). The CompLex corpus contains 1,800 occurrences of multi-word expressions in its three sub-corpora. A multi-word expression is represented by a Noun-Noun or Adjective-Noun pattern, followed by any POS tag that is not a noun. At the LCP-2021, the original CompLex data was augmented by increasing the number of annotations on the same data by utilizing Amazon's Mechanical Turk platform. This produced CompLex 1.0, which was used by teams. This was done to address the reliability problems in the original CompLex data set (Shardlow et al., 2020). For the first data set only instances that were marked as complex by four or more annotators were kept, the average number of annotations per instance being seven. Using the Mechanical Turk platform they requested a further ten annotations of each on the instances.

[7] Lexical Complexity Prediction 2021 data `https://sites.google.com/view/lcpsharedtask2021/call-for-participation`

2.6.3 Best-performing systems 2021

In 2021, it was evident that transformer-based models outperformed other models for LCP. This was especially true when a variety of transformers were used to create an ensemble-based model.Shardlow et al. (2021a)

Rank	Team	Pearson	Spearman	MAE	MSE	R2
1	JUST BLUE	0.7886	0.7369	0.0609	0.0062	0.6172
2	DeepBlueAI	0.7882	0.7425	0.0610	0.0061	0.6210
3	Alejandro Mosquera	0.7790	0.7355	0.0619	0.0064	0.6062
4	Andi	0.7782	0.7287	0.0637	0.0064	0.6036
5	CS-UM6P	0.7779	0.7366	0.0803	0.0100	0.3813
6	tuqa	0.7772	0.7344	0.0635	0.0068	0.5771
7	OCHADAI-KYOTO	0.7772	0.7313	0.0617	0.0065	0.6015
8	BigGreen	0.7749	0.7294	0.0629	0.0065	0.5983
9	CSECU-DSG	0.7716	0.7326	0.0632	0.0066	0.5909
10	ia pucp	0.7704	0.7361	0.0618	0.0066	0.5929

Table 2.8: Top 10 best-performing systems from LCP-2021 in terms of Pearson, Spearman, MAE, MSE, and R2, ranked in Pearson order (Shardlow et al., 2021b).

JUST BLUE

The best-performing JUST BLUE (Yaseen et al., 2021) used the pre-trained language models, BERT(Devlin et al., 2018) and RoBERTa (Liu et al., 2019b) models. They imported the BERT model and used a scikit-learn wrapper to fine tune BERT, as it includes SciBERT and BioBERT models for the scientific and biomedical fields. They used simple transformers and classification libraries to import the RoBERTa model. This system obtained the highest Pearson Correlation score of 0.788 using the pre-trained language models BERT and RoBERTa. This program utilized four language models: two BERT models and two RoBERTa models. Bert1 and RoBERTa1 were provided with individual target words, while Bert2 and RoBERTa2 received the corresponding sentences for each target word, allowing for context to be taken into account. These models predicted the complexity level of the inputted words or sentences, which was determined by a weighted average. Model 1 had a weight of 80% and Model 2 had a weight of 20%, indicating that the complexity of target words was given more importance than the complexity of other words in the sentence. However, the program still considered the context of each sentence when calculating the weighted average, as previous studies have shown that context can significantly affect complexity prediction (North et al., 2023).

DeepBlueAI

The DeepBlueAI (Pan et al., 2021) system performed exceptionally well in both subtasks, achieving the highest Pearson's Correlation for Sub-task 2 and the second highest for Sub-task 1. It also had the highest R2 score overall. To achieve these results, the system used a combination of pre-trained language models that were fine-tuned for the task using techniques such as Pseudo Labelling, Data Augmentation, Stacked Training

Models, and Multi-Sample Dropout. Transformer models were used to encode the data, with the genre and token serving as a query string and the given context as supplementary input. Pan et al. concluded their model's state-of-the-art performance in both sub-tasks was due to its use of multiple transformers and training strategies. This is congruent with the findings from the best-performing JUST BLUE (Yaseen et al., 2021) system that also used a diverse range of models.

Alejandro Mosquera

Mosquera's system illustrated that feature engineering is still a viable approach as this system was the third best-performing. It uses 51 features. From the pycholinguistic features, average age of acquisition (AOA): At what age the target word is most likely to enter someone's vocabulary, was found to be the most important. Additionally the system used a selection of lexical, contextual and semantic features at both target word and sentence level. It treated both sub-tasks as regression problems since the labels in the training data set were continuous. To address sub-task 1, they utilized the gradient tree boosting implementation of LightGBM (LGB) (Ke et al., 2017). They performed minimal hyper-parameter optimization with a 0.01 learning rate and restricted the number of leaves of each tree to 30 over 500 boosting iterations based on the development set. For sub-task 2, they obtained the complexity score of each MWE component using a linear regression (LR) model and averaged it with equal weights.

2.7 Summary of approaches for complexity prediction

Each of the best-performing systems from the three shared tasks performed well on their particular data set but each data set has its own characteristics. Therefore, when trying to establish the best-performing overall approach it is sensible to establish themes rather than a precise model. At CWI-2018, all of the top sytems CAMB(Gooding and Kochmar, 2018), NILC(Hartmann and Dos Santos, 2018) and ITEC (De Hertog and Tack, 2018) used MRC Database psycholinguistic features. After analyzing the systems that took part in the 2021 task, it was found that there was not much difference between the deep Learning and feature Based approaches.

After reviewing the three international competitions, it was found that various machine learning (ML) models have been used. These include Support Vector Machines (SVMs), Decision Trees (DTs), Random Forests (RFs), neural networks, and state-of-the-art transformers like BERT (Bidirectional Encoder Representations from Transformers), RoBERTa(Robustly optimized BERT approach), and ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately). Word embeddings that consider context, such as RoBERTA, were used by many top performing systems in the 2021 competition. According to North et al. (2023) ensemble-based models, which combine several of these models, were the best approach before the more recent emergence of transformer-based models. This section will provide a brief description of the various models used.

2.7.1 Support Vector Machines

Support Vector Machines (SVMs) are ideal for binary classification tasks. They deliver outstanding results when there is a clear separation between the two classes. However, SVMs are less effective when dealing with multiple classes or a large number of features since this can lessen the uniqueness of each class. Previously, SVMs were popular in early CWI research, which concentrated on predicting binary complexity. However, as the task has changed to LCP on a scale, their use has declined. Four systems (Malmasi et al., 2016; Sanjay et al., 2016; Kuru, 2016; Choubey and Pateria, 2016) used SVMs, as can be seen in the in the full list of systems submitted to SemEval-2016 in Table 2.1. At SemEval 2016, the AmritaCEN teamSanjay et al. (2016) used SVM with word embedding features, orthographic word features, similarity features and POS tag features. It is worth noting that their system w2vecSimPos scored worse out of the four systems and suggest that POS tags are less of an important feature for CWI. More recently, Yaseen et al. (2021) with their JUST BLUE system at CWI-2021 discovered that between SVM, RF, BERT, and RoBERTa models, along with a BERT and RoBERTa hybrid model, a BERT and RoBERTa hybrid model achieved the highest performance.

2.7.2 Decision Trees and Random Forests

Decision Trees (DTs) are a type of algorithm that use learned rules to make accurate predictions. They work by filtering labeled data through decision nodes or branches until the data is accurately separated based on class. They use the values in each feature to split the data set to a point where all data points that have the same class are grouped together DTs are known to outperform SVMs in the context of CWI, possibly because they are better equipped to handle features that overlap between classes. At CWI-2016 the most common and arguably the most successful CWI systems were either a DT or a RF model. As can be seen in Table 2.1 6 systems used this approach. RFs are comprised of multiple DTs, each trained on a random subset of the data. With limited input, each DT learns a sequence of hierarchical rules for classification. RFs generate their final output through a plurality voting system. Due to each DT only observing a small fraction of the data, RFs are less prone to overfitting. Each DT learns to distinguish its inputted classes without making sweeping generalizations across the data set. Consequently, each DT becomes specialized in identifying the distinguishing features of its limited input. By pooling these DTs together, an RF is more adaptable to unseen data than a stand-alone DT. Therefore, RFs are better suited to dealing with large data sets with numerous features compared to a single DT.

2.7.3 Ensemble-Based Models

Ensemble-based models consist of multiple sub-models that work together to produce a final output through a form of voting. These sub-models can be of the same type, like an RF, or different types. The diversity of ensemble-based models is their main advantage, as they can utilize the strengths of various models, such as SVMs, DTs, RFs, neural networks, or transformers while mitigating the disadvantages of relying on only one type of model. As a result, ensemble-based models are currently the best option for LCP. Over time, various combinations of sub-models have been used. However, according to a research paper by Zampieri et al. (2017), an ensemble classifier that used predictions from multiple systems in the 2016 task performed worse as more systems were added. They employed ensemble classifiers by utilizing the output of the 2016 SemEval systems. This method involves training several classifiers and merging them through ensembles. Figure 2.2 shows how plurality voting ensembles performed using the output of all systems, and the top-10 ranked systems showed the best performance.

However, the setup that utilized the top-10 systems performed well but still fell short of the best system in the competition. In summary, the best-performing system was the smallest ensemble. This research suggests that keeping it simple is a good heuristic to apply with ensemble-based models.

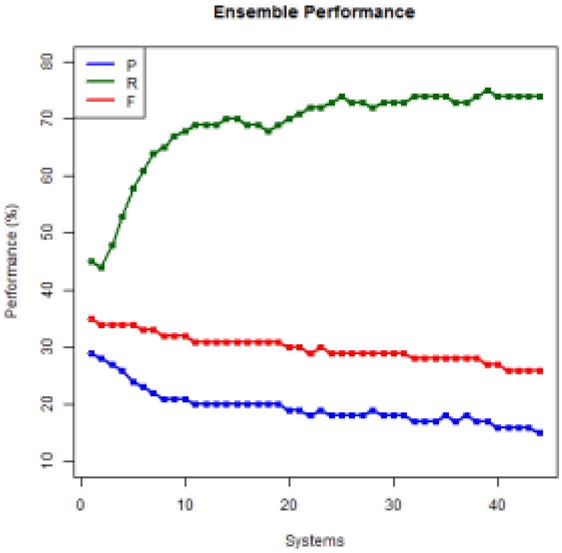


Figure 2.2: Ensemble system built from output of systems in SemEval 2016 Zampieri et al. (2017) **P**recision, **R**ecall and **F**1-score.

Chapter 3

Method

This chapter is divided into four sections. The first section describes a baseline system with features inspired by the baseline classifier that was used in the CWI-2018 shared task (Yimam et al., 2018). The second section shows the method for building a system inspired by the best-performing CAMB system(Gooding and Kochmar, 2018). The third section is a description of a CAMB_A model that investigates additional features from learner corpora data and contextual features from (Liu et al., 2019b). The last section is the final system with a list of the features chosen from those described in the previous sections. All systems and models are trained using training data and tested on the development sets that were used in the original 2018 task, apart from the final system that was tested on the test data. After investigating the impact of the added features, the final system is then run on the test data that was used in the original competition. All of the systems use the data from CWI-2018 in the original splits of training, development and test that were released to participants of the original competition. This is described in more detail in subsection 2.5.2. Although the data was left unchanged, it was further split into non-native and native annotations to compare the differences between these two groups. This was done to answer sub-question 2 *"To what extent do the complex word annotations by non-native and native speakers have an impact on the performance of the CWI models? Is this difference measurable?"*. Each system has a description of the resources that were used for the features and links to publicly available sources. As with the original task, the goal was to provide models that would perform with the binary and probabilistic data that was provided. Therefore, each system includes a description of the different methods used for these two classifications.

For the CAMB-influenced model, some resources were not freely available, and therefore the models described use only resources that are publicly available and do not require payment. For instance, the GoogleWeb 1T 5-Grams used for word frequency in the baseline system is only available from the Linguistic Data Consortium for a fee of $150. Similarly, the CALD (Cambridge Advanced Learner's Dictionary) used in the CAMB model to add CEFR-level information for the target words can be obtained in a straightforward manner by extracting the data through the site [1]. However, upon reviewing the documentation, this was found to be prohibited.

[1] https://www.englishprofile.org/wordlists/evp

3.1 Data Description

Total instances in the data set is shown in Table 3.1 along with the mean probabilistic complexity score.

File Name	Total Instances	Mean Probabilistic
WikiNews_Train	7745	0.42
WikiNews_Dev	869	0.41
WikiNews_Test	1286	0.42
Wikipedia_Train	5550	0.45
Wikipedia_Dev	693	0.49
Wikipedia_Test	869	0.5
News_Train	14001	0.4
News_Dev	1763	0.39
News_Test	2094	0.38

Table 3.1: Statistics for CWI-2018 data.

As shown in Table 3.2 more words are labelled as complex by the non-native speaker annotators than the native speakers. This seems anomalous because if English was the annotators' second language, then it would seem logical that they would find more instances of complex words and phrases. However, it makes sense in the context of the directions that were given to annotators to label words that they thought would be complex for language learners and not themselves.

File Name	Total native complex	Total non-native complex
WikiNews_Train	2446	2288
WikiNews_Dev	281	250
WikiNews_Test	400	389
Wikipedia_Train	1892	1785
Wikipedia_Dev	273	234
Wikipedia_Test	341	294
News_Train	4173	4152
News_Dev	529	509
News_Test	596	627

Table 3.2: Number of rows labeled as complex for Native and Non-Native data.

For the comparison of the classifiers with Non-Native and Native each Genre was split into separate sets. The data was split after the features had been extracted. The NATIVE data contains rows where 'Native' is not null and 'Non-Native' is either null or equal to 0. The NON-NATIVE data contains rows where 'Non-Native' is not null and 'Native' is either null or equal to 0. The function splits all files in a folder into two new subfolders one called "native" and one called "non-native" . The 'Native' and 'Non-Native' columns refer to columns 7 and 8 in Table 2.5. This was done to try and establish to what extent annotations by non-native and native speakers impact the performance of the CWI models and investigate the performance of different features on these two demographics.

3.2 Description of Baseline Systems

In the 2018 shared task competition, two simple baseline systems were developed for binary and probabilistic classification tasks. The baseline system only utilized the essential six frequency and length features described in (Yimam et al., 2017b). In summary, the baseline system uses the length of the word, the number of vowels, the number of syllables. It also has three corpora frequency features: the frequency of the word in Simple Wikipedia, the frequency of the word in the paragraph in the original paragraph shown in the (HIT) Human Intelligence Task, and the frequency of the word in the GoogleWeb 1T 5-Grams. The system employed the Nearest Centroid classifier and Linear Regression algorithms from scikit-learn for binary and probabilistic classification, respectively. The binary classification task was assessed using accuracy and macro-averaged F1 metrics, while the probabilistic classification task used the Mean Absolute Error (MAE) measure. The MAE calculates the average deviation between the values predicted by the system and the values in the gold standard for all the test instances, as detailed in section 2.2. For the probabilistic monolingual English track the baseline system performed better than approximately 50% of the systems in the competition as seen in Table 2.7. It scored an MAE of 0.1127, 0.1053 and 0.1112 for the NEWS, WIKINEWS and WIKIPEDIA data respectively. This baseline system would have ranked 9th for NEWS and WIKINEWS data and 10th for WIKIPEDIA data if it had been in the competition. For the binary classification, the baseline systems were more widely outperformed. Out of the 39 systems that were entered for the NEWS data only 5 scored worse than the baseline. This fall to 3 for WIKINEWS and 8 for the WIKIPEDIA data. For the binary track approximately twice as many systems were entered with many teams entering multiple systems for each data set. The baseline system performed best on the NEWS data scoring an F-1 of 0.7579 F-1. For the WIKINEWS and WIKIPEDIA data it scored lower with 0.7106 and 0.7179, as shown in Table 2.6.

3.2.1 Prepossessing

For all of the systems and features that were investigated, the same preprocessing steps were carried out. The cleaning operations that were done to the 'sentence' column as shown in Table 2.5. These included removing punctuation and replacing "%" with "percent". The 'Target' column was split into separate words using white space and storing the result in a new column called 'split'. The number of words in is stored in a new column called 'count'. The WIKINEWS data has a difference from the other two genres, as can be seen in the following example:***[#37-1 Guatemalan Supreme Court approves impeachment of President Molina Yesterday in Guatemala, the Supreme Court approved the attorney general's request to impeach President Otto Pérez Molina.]*** The "#37-1" is added to the beginning of the sentence. This number is not present in the other two genres and will need to be removed in the preprocessing. Apart from this difference, the rest of the format is identical across the three genres. Therefore, the same preprocessing steps can be applied. This example sentence repeats over 24 rows, and 8 words are marked as complex. The exact word in the same place in the sentence is only repeated in the row if it is part of an MWE. The data is given in a tab-separated format and is in the format shown in Table 2.5 but without the column headings.

3.2.2 Frequency features

For the baseline system described in (Yimam et al., 2017a) the following frequency features were used: Simple Wikipedia, the word in the paragraph of the HIT (Human Intelligence Task), and frequency in GoogleWeb 1T 5-Grams (Brants and Franz, 2006). The Google 1T 5-GRam data can be obtained at `https://catalog.ldc.upenn.edu/LDC2006T13`. It consists of observed frequency counts of English word n-grams for around 1 trillion tokens collected from Web Text. However, this data is not freely available; therefore, an alternative is needed. Instead of this corpus, the Google Ngram data was used to find word frequencies. This data is freely available, but the corpus is made up of scanned books and not web text, so the performance is likely to suffer given the different genres of the data. However, the best performing CAMB system during the shared task in 2018 used Google Ngram data via the DataMuse API; therefore, it was evaluated to be sufficient for the baseline system. Three lexical features were used for the baseline system. These were the number of vowels per word or MWEs, the number of characters to give the length and the number of syllables. These features were extracted using the Datamuse API[2].

Google Ngram frequency data extraction

Two methods were investigated for obtaining the Google Ngram data, as the research indicates that word frequency is a crucial feature for the task Wróbel (2016); Shardlow (2013). The Google Books Ngram Corpus is a compendium of literary works that exclusively encompasses books. It does not include other forms of literature such as periodicals, websites, or spoken language. Each edition of a book is represented only once in the corpus. The corpus predominantly comprises books held in a select number of major university libraries, with over 40 included in Version 1. The entire Google Books Ngram Corpus Version 3[2] is huge; for British English words for all years, there are just under 2 trillion words.

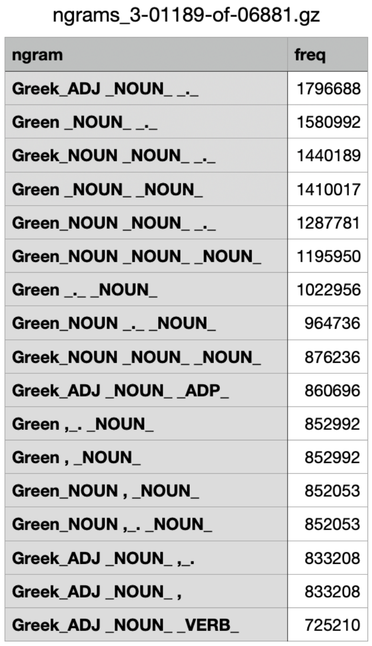
The first method was done by writing a function to download the raw data from the Google Ngram site. When creating the raw Ngram data, each file is opened in turn, and all unique Ngrams with a minimum frequency threshold of 1000 are collected. This resulted in a large number of raw data files. These files increased in size resulting in 421 2-grams, 5243 3-grams, 4330 4-grams and 10772 5-grams. An example of the raw 3-gram data is shown in Figure 3.1; the raw data contains POS information which is incorporated Ngram column. This large amount of data for the larger Ngrams takes quite some time to download. Due to the size of the data for the entire corpus, raw data was extracted for a period of ten years. The total number of words for 2010-2019 is over 45 billion. This made the downloading of the entire corpus impractical. After the extraction process, the downloaded files were removed to optimize computational resources. It is worth noting that, even with default settings in place, the top n-gram listings per .gz-file still necessitate approximately 36GB of storage space. Version 2 of Google Ngram data incorporated syntactic annotations that tag words with their part-of-speech and head modifier relationship (Lin et al., 2012). The corpus consists of over 8 million books, which accounts for 6% of all books ever published. However, working with these raw data files ultimately proved to be too computationally cumbersome. Furthermore, it did not produce results for all target multi-word expressions. Therefore,

[2] https://www.datamuse.com/api/

[2] https://storage.googleapis.com/books/ngrams/books/datasetsv3.html

this method was not used in the final systems. The alternative approach using the API was used described in the next paragraph.

The second method investigated for Ngram frequency was using an API[1]. This API provides information on Ngrams and returns both an absolute total match count and a relative total match count. The former is the total sum of absolute match counts based on the year, while the latter is the absolute total match count divided by the absolute total match count of all Ngrams of the same length or probability. After a comparison of the results for both methods, it was decided to use the second method, which uses the Ngram API. This was because the first method that used raw data was very slow, and the performance was not considerably better. Both systems were not able to return frequency data for all MWEs. Both methods failed to have 100% coverage of all target multi-word expressions. The total match count and a relative total match count were both used as separate features that were input for the system as separate features. This method was much more time efficient, which offset the potential gains from the first method using the raw Ngram data. Ideally, methods would be combined to enable the maximum number of Ngrams to have frequency values for as many MWEs as possible. However, this was not able to be achieved.



ngrams_3-01189-of-06881.gz

ngram	freq
Greek_ADJ _NOUN_ _._	1796688
Green _NOUN_ _._	1580992
Greek_NOUN _NOUN_ _._	1440189
Green _NOUN_ _NOUN_	1410017
Green_NOUN _NOUN_ _._	1287781
Green_NOUN _NOUN_ _NOUN_	1195950
Green _._ _NOUN_	1022956
Green_NOUN _._ _NOUN_	964736
Greek_NOUN _NOUN_ _NOUN_	876236
Greek_ADJ _NOUN_ _ADP_	860696
Green ,_. _NOUN_	852992
Green , _NOUN_	852992
Green_NOUN , _NOUN_	852053
Green_NOUN ,_. _NOUN_	852053
Greek_ADJ _NOUN_ ,_.	833208
Greek_ADJ _NOUN_ ,	833208
Greek_ADJ _NOUN_ _VERB_	725210

Figure 3.1: Raw 3-gram data example

Simple Wikipedia Frequency (simple_wiki)

For this feature, the simple_wiki.txt file was created, including the simplified version of 10,000 sentences. The target word was then searched for, and the frequency was counted. Historically, data sets have utilized Simple Wikipedia and edit histories as the primary means for annotating complex words and are viewed as a "gold standard". However, there has been considerable debate surrounding the appropriateness of using

[1] https://github.com/ngrams-dev/general/wiki

Simple Wikipedia for text simplification, as evidenced by numerous studies (Amancio and Specia, 2014). This data is freely available because it is from Wikipedia but can be obtained from `https://cs.pomona.edu/~dkauchak/simplification/` in a readily usable format.

Frequency in original paragraph (HIT_freq)

The last feature for the baseline was the frequency in the original paragraph displayed to annotators.This feature simply counts to see how many times the target word is in the original paragraph of text that was used for the HIT on Amazon Turk. The function checks the word in all sentences with the same ID, as each paragraph displayed to annotators was given a unique ID in the data.

3.2.3 Description of Classification Algorithms

Both models make use of Sklearn(Pedregosa et al., 2011) for the classification tasks. To train the models, the Nearest Centroid and Linear Regression algorithms were used for binary and probabilistic, respectively. These were chosen as they were used by the best-performing model at CWI-2018. Furthermore, the research showed that the type of classification algorithms used in 2018 did not hugely affect performance with the feature based systems.

Binary Classification

To make predictions for new examples, the Nearest Centroid or nearest prototype classifier summarizes the training data set into a set of centroids (centers).This classifier work in a similiar way to K-Nearest Neighbours classifier. During training, the centroid is computed for each target class. Once training is complete, if a point (let's call it "X") is given, the distance between X and each class's centroid is calculated. The minimum distance is then selected from all the calculated distances. The class to which the centroid of the given point's minimum distance belongs is assigned to that point.

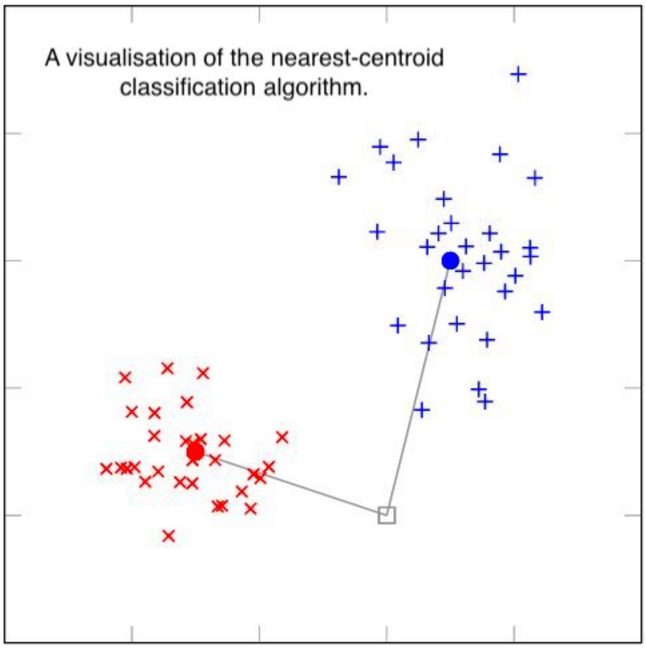


Figure 3.2: Visualisation of Nearest Centroid Classifier

In the visualisation of the Nearest Centroid classifier in Figure 3.2, a training data set has already been clustered into two classes. To classify input data, the distance is compared in the feature space from the mean position of each cluster. The pluses and crosses indicate the vector positions of different instances of data in the feature space, while the filled circles signify the cluster mean positions. The square represents a vector that still requires classification.

Probabilistic Classification

For the probabilistic classification, identical features are used, and the Linear Regression algorithm is used to estimate target values from the sklearn.linear_model. Linear regression involves finding the best fitting straight line for a set of scattered data points, and so is well suited to the task. Multiple linear regression classifiers are trained on different sets of features extracted from each genre.

3.3 CAMB Model description

The CAMB model was re-implemented using only corpora that were publicly available. The system creates a model for the classification of single words as used in the baseline system. If there are multiple words in the word column of the data, the system utilizes three binary classification methods and one probabilistic classification method to forecast the complexity of phrases. The three binary classification methods for phrases are: the individual words in the phrase are classified using the single complex word classifier, and if the total number of complex words is above a ”pre-defined threshold”, then the phrase is marked as complex. There is no detail in the paper on what the threshold was so for this system. I set the threshold to one. The second for MWEs was an ngram classifier. The frequency of n-grams contained within phrases is obtained from the Corpus of Contemporary American English (COCA) (Davies, 2009). COCA is not entirely free; while some functionality is freely available, full access requires a paid subscription. To classify new phrase instances, I train an AdaBoost classifier using these frequencies as features. Lastly, the system uses the Greedy algorithm approach and labels all MWEs as complex.

3.3.1 Lexicon-Based Features

These following lexicons were used in the CAMB system and are publicly available, and were used as features. The target phrase was searched for in each lexicon, and a value was returned if the word was present. The four lexicons used were as follows: 1) Ogden's Basic English[1] (Ogden and Halász, 1935): a list of 850 words from Ogden's Basic English list. The idea from Ogden was that 90% of the concepts in English are covered with this simple list. 2) SubIMDB: a list produced using the SubIMDB[2] corpus (Paetzold and Specia, 2016a). The SubIMDB corpus is a large structured corpus of subtitles of movies and series. The word frequency in the subtitles of films (Compiled Corpus of Movies and Series for Children). 3)Simple Wikipedia (SimpWiki): a list of the top 6,368 words contained in the Simple Wikipedia data. (Coster and Kauchak, 2011). And 4) Word Complexity Lexicon(Maddela and Xu, 2018) A human-rated word complexity lexicon of 15,000 English words. The complexity score of each word is determined by combining ratings from several people. The scores range from 1 to 6, with 1 being very simple and 6 being very complex. The ratings are based on individual assessments from 11 annotators for each word in the lexicon. A rating of -1 means that the annotator did not provide a score. Lastly, as with the baseline system, the sentence length is counted along with the number of words in the sentence where the target word occurs.

3.3.2 Lexical Features

These were extracted for the baseline system and were used unchanged for CAMB-inspired re-implementation. They can be summarised as: 1)The word length: the number of characters in the word. 2) The number of syllables: the syllable count for the target word. 3) The WordNet Features: number of synonyms, number of hypernyms and hyponyms for the word's lemma from WordNet, and lastly, 4) POS tags were extracted using Stanford Core NLP.(Manning et al., 2014).

3.3.3 Description of MRC Psycholinguistic Database Features

Some adjustments were needed to collect this corpus because the site is limited to a 5000-word output. The full corpus used in this thesis was achieved by filtering the output by word length and then using multiple enquiries to achieve a complete list.

The following nine features were collected: The number of phonemes (NPHN) references how many phonemes are in the target word. These are the 44 sounds that make up every English word. The next three features originally come from the Brown Corpus (Kucera and Francis, 1967). The Kucera-Francis written frequency (KFFRQ),

[1] http://ogden.basic-english.org/

[2] http://ghpaetzold.github.io/subimdb/

Kucera-Francis number of categories (KFCAT) and Kucera-Francis number of samples (KFSMP) are derived from the classic work "Computational Analysis of Present-Day American English" (Maverick, 1969), which provided basic statistics on what is known today simply as the Brown Corpus. The Brown Corpus is divided into 500 samples of 2000+ words each. These numbers refer to the total number of occurrences in the corpus and the total number of occurrences in categories and samples. A full list of the 500 samples and categories is available in the Brown Corpus Manual (Francis and Kucera, 1979).

The Thorndike-Lorge written frequency (T-LFRQ) is a measure of how often English words appear in representative general reading material. This information is taken from *The Teacher's Word Book of 30,000 Words* (Thorndike and Lorge, 1944), which lists words that occur at least once per million words. Each entry in the book is alphabetized and includes five columns of data: occurrences per million words, the Thorndike general count from 1931, the Lorge magazine count, the Thorndike count from 120 juvenile books, and the Lorge-Thorndike semantic count used in MRC Psycholinguistic data.

The Familiarity rating ranging (FAM), Concreteness rating (CNC)and Imageability rating (IMG) give values from 100-700 and are made from a combination of three scales (Gilhooly and Logie, 1980; Toglia and Battig, 1978; Paivio et al., 1968). Lastly, the Age of Acquisition (AOA) (Gilhooly and Logie, 1980) measures for 1,944 words where subjects rated words on a scale that ranged from 1 (age 0-2 years) to 7 (age 13 years and older) with 2-year sub-divisions. This number is then multiplied by 100, also giving a score from 100-700.

3.3.4 Summary of Feature extraction process

For the binary shared task, CAMB's submission uses a straightforward greedy method for phrase classification. This was also done of the re-implemented version. The word features are populated by getting the syllables and word length for each word using the Datamuse API. The syatem then parses sentences using StanfordCoreNLP It extracts the relevant linguistic features (POS, dependency, lemma, etc.). Next, the system obtains MRC features (AOA, CNC, IMG, etc.) for each word using the extracted MRC corpus. The sysstem then gets the additional linguistic features like synonyms, hypernyms, and hyponyms using WordNet. After that it checks if the words are present in specific word sets (ogden, simple_wiki, sub_imdb) and adds binary features accordingly. Lastly, it gives the word frequency from Google using the Datamuse API.

3.4 CAMB_A Feature Description

Following is a description of features investigated additionally to the CAMB-inspired re-implementation described in the previous section. All features were added to the existing set of features from the previous section. The features described were added individually, and their individual impact is as shown in the Results section 4.3.

3.4.1 RoBERTa Embeddings

RoBERTa is a self-supervised transformers model pretrained on raw English texts, using publicly available data to generate inputs and labels. To obtain contextualised word embeddings, the RoBERTa base model was used (Liu et al., 2019a) in combination with PyTorch (Paszke et al., 2019). Initially, each word in the target sentence and the target word or phrase had embeddings extracted, and then the target phrase or word was matched against the embedding in the target sentence. However, this method was insufficient as the embeddings were often different for the target phrase embedding as for the word in the sentence, as the embedding changed based on the context. What was remarkable was how many target words had the same embedding when they were processed in isolation and when they were processed as part of the target sentence. However, this mistake was rectified to make sure the target word embedding was captured from the sentence and therefore contained the correct contextual information. The final script extracts the word embeddings of the target word or MWEs in the context of the given sentence by using the start and end index supplied in the original data. The start index and end index variables are found in the third and fourth columns, as shown in Table 2.5. The numbers are used to find the position of the target word in the sentence and return the contextualised embedding. The return-offsets-mapping parameter that enables this is only available for tokenisers that are derived from transformers. PreTrainedTokenizerFast, which is available from Hugging Face [3]. Therefore, RobertaTokenizerFast was used instead of the RobertaTokenizer to use this feature. Lastly, the RoBERTa embedding returned an array which was flattened. By flattening the embedding, there were an additional 763 feature columns added. This way, the model does not see the ”Embedding” column as a single feature but rather sees each dimension of the embedding as a separate feature. This allows for compatibility with the scikit-learn API and models. The embedding extraction and handling of this large dataframe were done using Google Colab as extracting the embedding was computationally expensive. To handle this multi-column data BaseEstimator, TransformerMixin were used from Sklearn (Buitinck et al., 2013) when the model was trained using these features. These features were then incorporated into the feature union pipeline to include these new columns as with the existing features.

3.5 Learner Corpora features

Lang-8 Learner Corpus

The Lang-8 corpus (Mizumoto et al., 2012) is correction data of learner English from Lang-8 [4], a site for language learning. The data was crawled in September 2011, and it can be accessed at `https://sites.google.com/site/naistlang8corpora`. First, the data is filtered only to include learners of English. The first feature used for the Lang-8 data was if the English language learner used the target word. This intuition is that learners will not use a complex word if they are of quite a low level. The learner data reviewed in this corpus was mainly A1 -B2 level English. The feature is the number of times learners use the target word. However, this feature does not reflect whether the word was used correctly, merely that the learner used it. The Lang-8 corpus was used at the NAACL HLT 2018 shared task by the TMU team (Kajiwara and Komachi, 2018) and ranked 6th for the NEWS genre data set and had the best-performing system for the NEWS data in the probabilistic classification results as can be seen in Table 2.7.

[3] `https://huggingface.co/`

[4] `http://lang-8.com`

The EF-Cambridge Open Language Database (EFCamDat)

EFCAMDAT is a learner corpus featuring essays from adult learners of English around the world. EFCAMDAT currently contains over 83 million words from 1 million assignments written by 174,000 learners across a wide range of levels (CEFR stages A1-C2)Geertzen et al. (2013). This corpus was created by refining and modifying the largest open-access L2 English learner database – the EFCAMDAT. The processed EF-camdat[5] (Shatz, 2020) removed common markup tags and texts with excessive or varied markup tags. Additionally, they removed ultra-short texts with less than 20 words and those containing a significant amount of non-English writing. Duplicate material found in other texts and outlier word counts (i.e. extremely high or low) were also removed from the sample. Furthermore, they identified the prompt for each text and split them into two data sets based on whether they were written in response to the original or second prompt. For use in the CAMB_A system, the data was split into five sub-files based on the CEFR level; there was no data for the C2 level, hence only five files. The target word or phrase was then searched for, and the number of times it occurred for the level was returned and added to the existing features. This resulted in a total of five additional features, one for each CEFR level.

3.5.1 Final System Features

The final system utilized features from the CAMB-inspired model but excluded those from the CAMB_A investigation. The features included syllables, length, dep num, synonyms, hypernyms, Ogden, simple_wiki, CNC, IMG, sub_imdb, google frequency, KFCAT, FAM, KFSMP, KFFRQ, AOA, NPHN, and T-LFRQ. For the binary classification, the system used the Adaboost algorithm and Logistic regression for the probabilistic, as described in the CAMB-inspired section. These features were chosen based on analyzing the results of the previously described models that are shown in the next section.

[5] `https://corpus.mml.cam.ac.uk/`

Chapter 4

Results

As in the original CWI-2018 shared task, the tests were run on the WIKIPEDIA, NEWS and WIKINEWS data set separately. This was done in order to make a comparison with the systems in the original task. The binary systems are evaluated based on their accuracy, precision, recall, and F1-score (Equation 2.1) and the probabilistic systems use the Mean Absolute Error (MAE) (Equation 2.3. The evaluation metrics that were used in CWI-2018 are explained in the chapter on Related Work 2.2.

4.1 Binary Baseline Models

The features used to obtain these results were: the number of syllables, length of the word, number of vowels, frequency in the simple Wikipedia corpus, frequency in the original HIT paragraph and Google frequency, as described in the Method chapter 3.2.

The Tables 4.1, 4.2 and 4.3 show the results of classifiers trained on each genre and then tested on each of the development data sets. The "All-data" model was trained on the combined data. Table 4.4 shows the same classifiers tested on the combined development data from all three genres.

Model Name	Accuracy	Precision	Recall	F-Score
Wikipedia	0.645	**0.593**	0.864	0.704
WikiNews	**0.646**	0.592	**0.885**	**0.709**
News	**0.646**	0.592	**0.885**	**0.709**
All-data	0.645	0.591	0.882	0.708
2018 Baseline	-	-	-	0.718

Table 4.1: Results of classifiers tested on WIKIPEDIA development data set.

In Table 4.1 it can be seen that there is no difference in performance on the WIKIPEDIA data between the models that were trained on WIKINEWS and NEWS training data, and these models attained the highest F-Score on across all of the baseline models that were tested separately on each genre.

For the baseline models, it was expected that the All-data model trained on all the data would perform best as it had a larger and more diverse amount of data to train on. However, this was not the case, with the All-data model only performing best when tested on the NEWS data. When tested on all the development data, the WikiNews-trained classifier outperformed the All-data model with an F-1 score of 0.639 compared to the All-data, which scored an F1-score of 0.669. None of the simple baseline classifiers

Model Name	Accuracy	Precision	Recall	F-Score
Wikipedia	**0.623**	**0.521**	0.835	0.642
WikiNews	0.613	0.514	**0.849**	0.640
News	0.612	0.513	0.847	**0.639**
All-data	0.610	0.511	0.841	0.636
2018 Baseline	-	-	-	0.710

Table 4.2: Results of classifiers tested on WIKINEWS development data set.

Model Name	Accuracy	Precision	Recall	F-Score
Wikipedia	0.639	0.524	0.908	0.665
WikiNews	0.635	0.521	0.920	0.665
News	0.635	0.521	0.920	0.665
All-data	**0.638**	**0.523**	**0.920**	**0.667**
2018 Baseline	-	-	-	0.758

Table 4.3: Results of classifiers tested on NEWS development data set.

Model Name	Accuracy	Precision	Recall	F-Score
Wikipedia	**0.636**	**0.539**	0.879	0.668
WikiNews	0.631	0.535	**0.893**	**0.669**
News	0.632	0.535	0.893	0.669
All data	0.632	0.536	0.890	0.669

Table 4.4: Results of classifiers tested on COMBINED development data.

performed the best on the data that they were trained on. These baseline systems all performed worse than the baseline system at CWI-2018, which scored an F1 of 0.7579 for NEWS, 0.7106 for WIKINEWS and 0.7179 for WIKIPEDIA, as seen in the binary classification results from CWI-2018 in Table 2.6. The confusion matrices for the baseline models are shown in Figure 4.1 Figure 4.2, Figure 4.2 and Figure 4.4. It is clear from the confusion matrices that all systems exhibit similar errors, particularly for positive labels in the test data predicted as negative by the models. Overall, the baseline classifiers all have very comparable or identical performance.

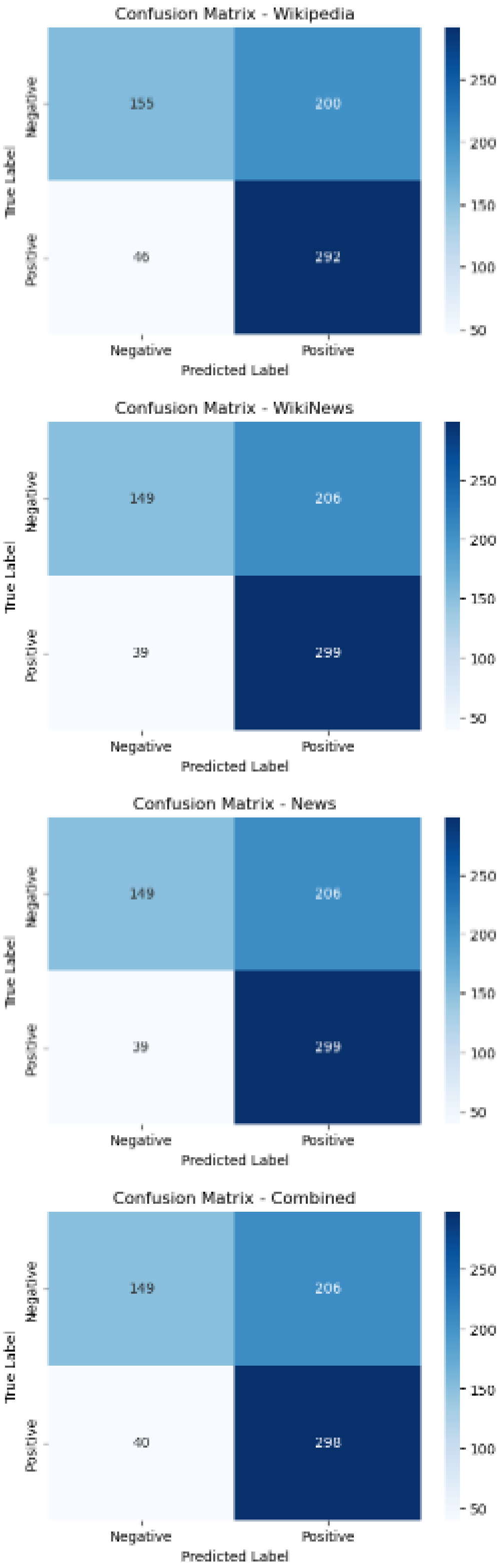


Figure 4.1: Confusion matrices for the baseline model ,trained on combined data sets, and the genre specific trained models. **Models tested on WIKIPEDIA Development data**. The total number of labels in WIKIPEDIA development data was 693.

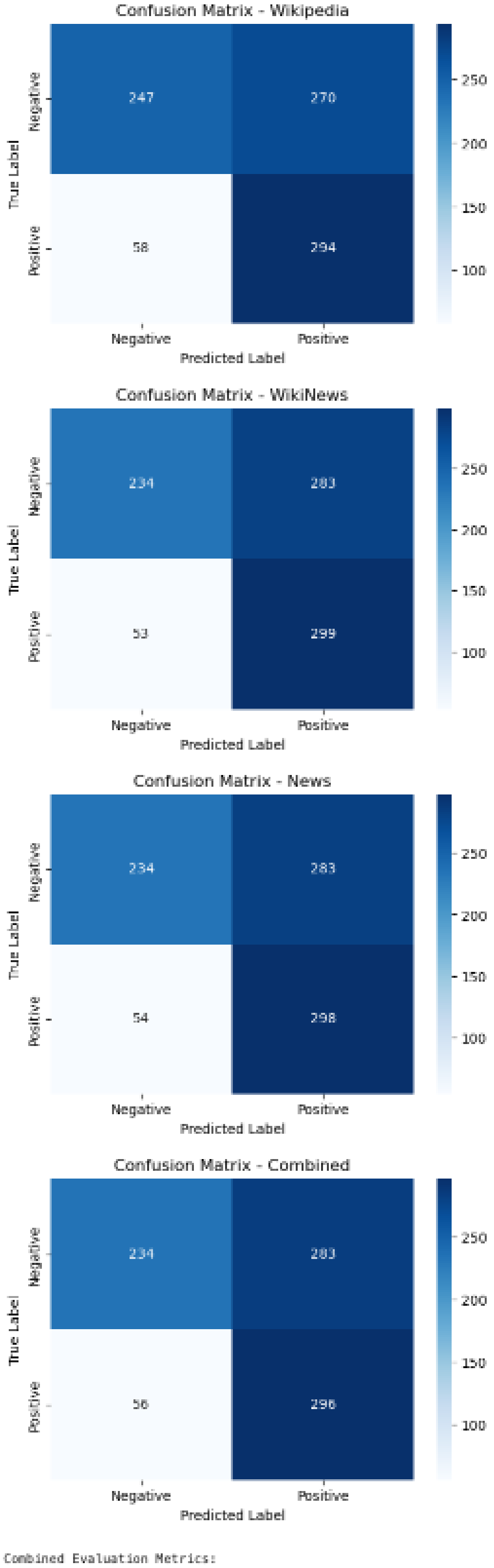


Figure 4.2: Confusion matrices for the baseline model ,trained on combined data sets, and the genre specific trained models. **Models tested on WIKINEWS development data.** The total number of labels in the WIKINEWS development data was 869.

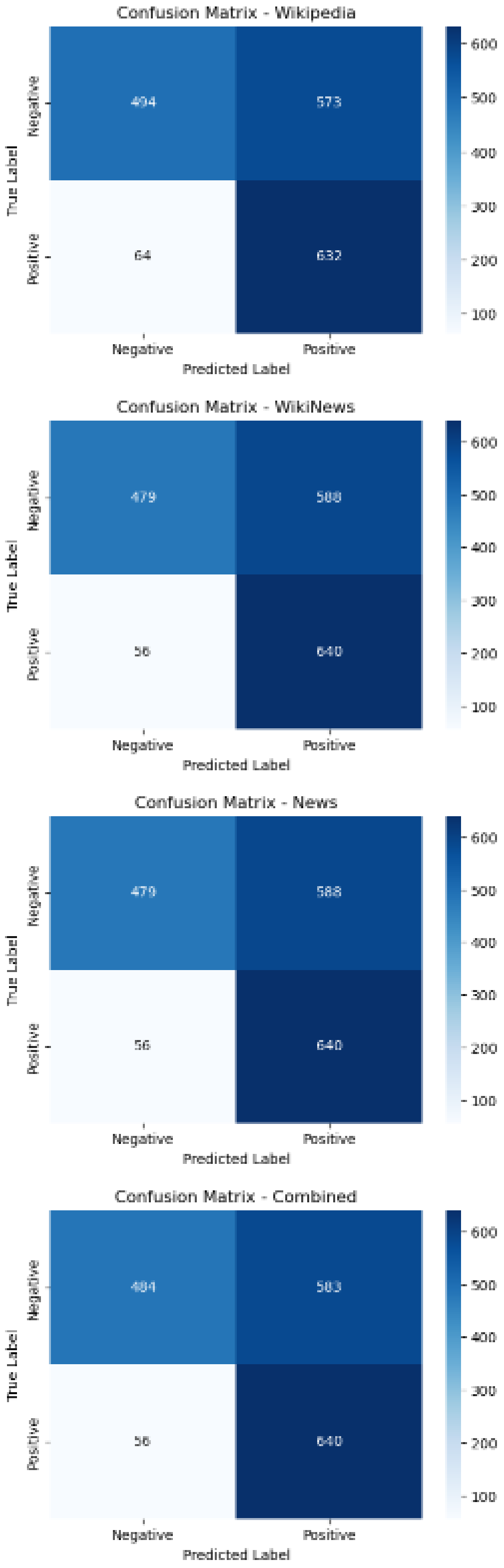


Figure 4.3: Confusion matrices for the baseline model ,trained on combined data sets, and the genre specific trained models. **Models tested on NEWS development data**. The total number of labels in the NEWS development data was 1763.

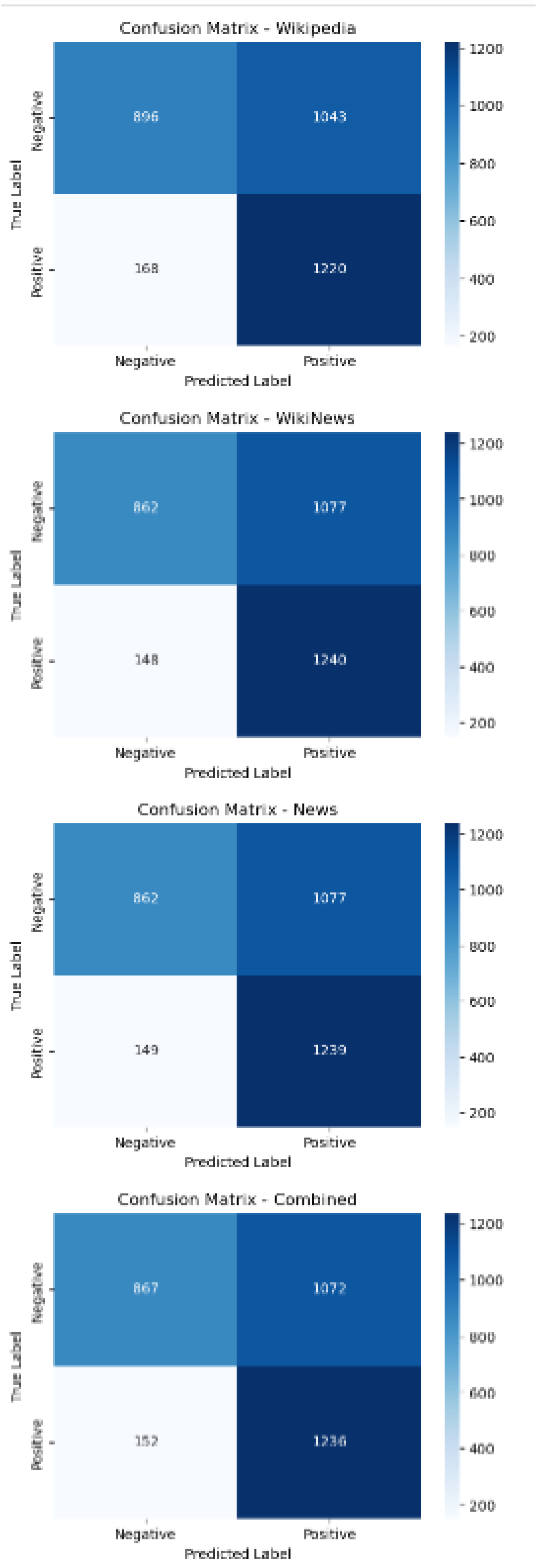


Figure 4.4: Confusion matrices for the baseline model ,trained on combined data sets, and the genre specific trained models. **Models tested on COMBINED development data**.

4.1.1 Probabilistic Baseline Models

The results for the probabilistic complexity prediction are shown in Table 4.5. The models use the same simple features of syllables, length, vowels, simple_wiki_freq, 'HIT_count' and Google frequency as used for the binary systems, the same features as for the binary classification. These results are comparable to the probabilistic baseline system used in the shared task. As with the binary models, the News-trained baseline model scored best across all test sets apart from the WIKIPEDIA data, where the All-data trained model beat it.

Model Name	WIKIPEDIA	WIKINEWS	NEWS	ALL DATA
News	0.1180	0.1136	0.1364	0.1267
WikiNews	0.1178	0.1115	0.1497	0.1331
Wikipedia	0.1192	0.1135	0.1527	0.1355
All-Data	0.1128	0.1128	0.1418	0.1293
2018 Baseline	0.1112	0.1053	0.1127	-

Table 4.5: Mean Absolute Error (MAE) per model - Tested all three data sets individually and all genres combined.

4.2 Results for CAMB-inspired system

4.2.1 Initial results for binary single word classifier

Table 4.6 shows the first results with a model called MYCAMB This model was trained and tested on single word data only. This data had all of the MWEs removed from training and the development set that was used for testing. The model named MYCAMB MWEs had the the full data set for testing and training and set all of the MWEs to be classified as complex.

Model Name	Precision	Recall	F-Score
MYCAMB Single word	0.800	0.785	0.793
MYCAMB MWEs (Greedy)*	0.809	0.796	0.802
Original 2018 CAMB	-	-	0.873

Table 4.6: New CAMB Model Trained on all train data and tested on all Dev (MWEs removed)*Result for training on NEWS Train and Testing on NEWS Dev.

4.2.2 Results for CAMB-inspired model including MWEs

Table 4.7 presents performance metrics of two models, the CAMB influenced model Without any learner corpus features and the scores for the original CAMB model from 2018 that used CEFR level in the Cambridge Advanced Learners Dictionary (CALD) as a feature. The models were evaluated on a data set containing Multi-Word Expressions (MWEs) using Greedy classification, where all MWEs were classified as complex. The results show that the original CAMB model outperformed the new CAMB model.

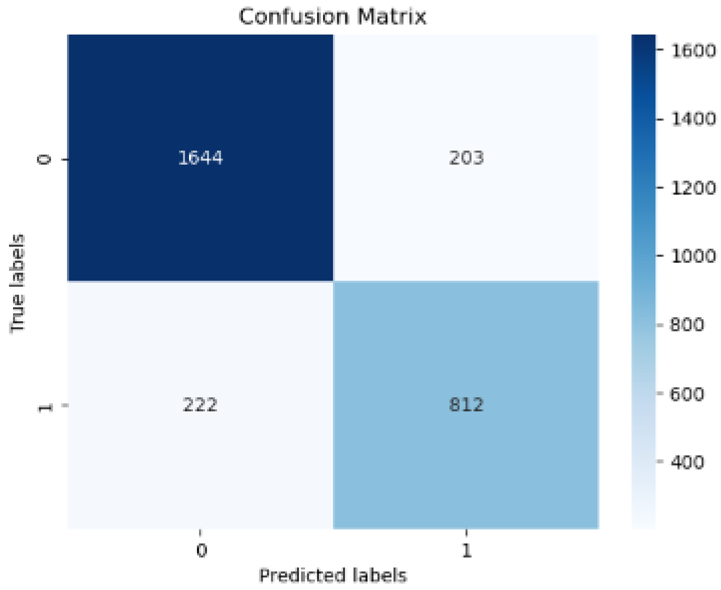


Figure 4.5: Single word CAMB classifier results trained on all single word train data and tested on all dev data.

Model Name	Precision	Recall	F-Score
MYCAMB (Without learner corpora)	0.841	0.820	0.830

Table 4.7: New CAMB Model Trained on all train data and tested on all Dev data with MWEs using Greedy classification.

4.2.3 Results for Genre specific classifiers

The results in Tables 4.8, 4.9 4.10 and 4.11 are for the CAMB influenced system using the original system's features without the any learner corpora feature. The MWEs were set to Greedy and all set to complex. It is noticeable that the model trained on WIKIPEDIA data does not perform better on WIKIPEDIA than the model trained on all data. The F-Score for the model trained on all the data performed best across all the genres.

Model trained on	**Tested on WIKIPEDIA**		
	Precision	**Recall**	**F-Score**
WIKINEWS	0.826	0.770	0.797
NEWS	0.833	0.767	0.799
WIKIPEDIA	0.821	0.784	0.802
All Data	0.843	0.821	**0.832**

Table 4.8: Results for Genre specific Model tested on WIKIPEDIA data.

Model trained on	**Tested on WIKINEWS**		
	Precision	**Recall**	**F-Score**
WIKINEWS	0.820	0.826	0.823
NEWS	0.831	0.802	0.816
WIKIPEDIA	0.766	0.824	0.794
All Data	0.861	0.842	**0.851**

Table 4.9: Results for Genre specific Model tested on WIKINEWS data.

Model trained on	**Tested on NEWS**		
	Precision	**Recall**	**F-Score**
WIKINEWS	0.800	0.866	0.832
NEWS	0.876	0.866	0.871
WIKIPEDIA	0.751	0.864	0.803
All Data	0.874	0.875	**0.874**

Table 4.10: Results for Genre specific Model tested on NEWS data.

Model trained on	**Tested on combined data**		
	Precision	**Recall**	**F-Score**
WIKINEWS	0.804	0.858	0.830
NEWS	0.867	0.853	0.860
WIKIPEDIA	0.754	0.856	0.802
All Data	0.871	0.868	**0.870**

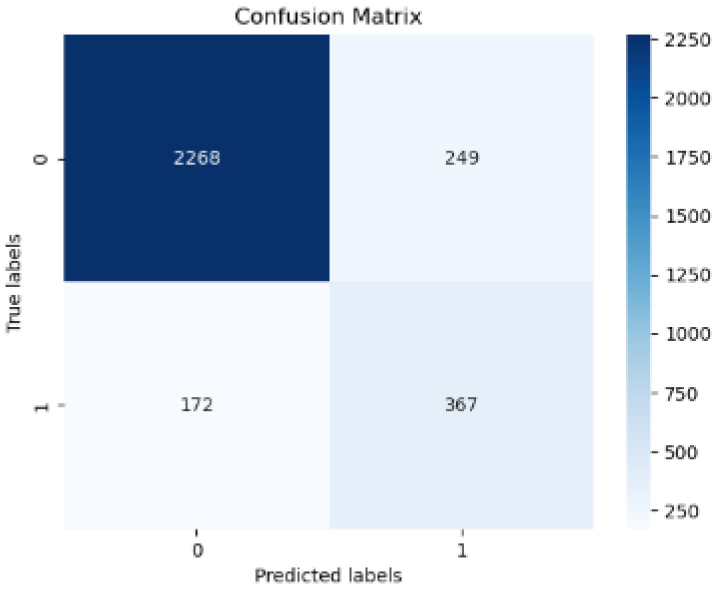
Table 4.11: Results for Genre specific Model tested on ALL data.

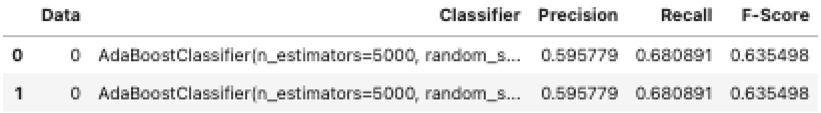
4.2.4 Non-native and Native Results

The CAMB model performs significantly worse on both data sets when the testing data is split into non-native and native, as shown in Table 4.12. These are for results using a classifier trained on all data. The Non-native Dev sets used for testing contained a total of 588 labels for WIKINEWES, 1254 labels for NEWS and 420 for WIKIPEDIA. The native set has 619 for WIKINEWS, 1254 for NEWS and 459 for WIKIPEDIA. Tables 4.13 and Table 4.14 show results for classifiers trained on Non-native and native data and then tested on each split set of data. The confusion matrices are shown in Figure 4.7 and 4.6.

Data Set	Precision	Recall	F-Score
NATIVE	0.622	0.670	0.645
NON-NATIVE	0.596	0.681	0.635

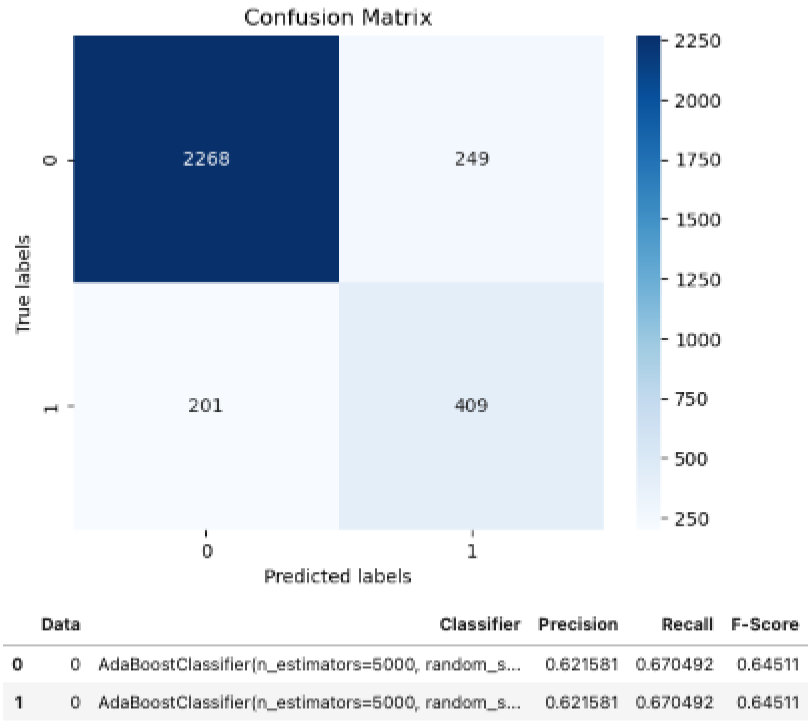
Table 4.12: New CAMB model results for Native and Non-native data splits with MWEs using Greedy classification.





	Data	Classifier	Precision	Recall	F-Score
0	0	AdaBoostClassifier(n_estimators=5000, random_s...	0.595779	0.680891	0.635498
1	0	AdaBoostClassifier(n_estimators=5000, random_s...	0.595779	0.680891	0.635498

Figure 4.6: Results for new CAMB tested on Non-native speaker data.



	Data	Classifier	Precision	Recall	F-Score
0	0	AdaBoostClassifier(n_estimators=5000, random_s...	0.621581	0.670492	0.64511
1	0	AdaBoostClassifier(n_estimators=5000, random_s...	0.621581	0.670492	0.64511

Figure 4.7: Results for new CAMB tested on Native speaker data.

Data Set	Precision	Recall	F-Score
NATIVE	0.855	0.695	0.767
NON-NATIVE	0.802	0.542	0.647

Table 4.13: Results for new CAMB trained only on Non-Native training data.

Data Set	Precision	Recall	F-Score
NATIVE	0.793	0.636	0.706
NON-NATIVE	0.784	0.681	0.729

Table 4.14: Results for new CAMB trained only on Native training data.

4.2.5 Probabilistic CAMB-inspired results

Table 4.15 provides a comparison of mean absolute error values for different models and their performance on different datasets using the Logistic Regression classifier. Lower mean absolute error values generally indicate better model performance, as they indicate that the model's predictions are closer to the ground truth values. When compared with the Original CAMB model that was submitted in 2018, it is clear to see that the models produced perform significantly worse. It is noticeable that in the original CAMB results, their model performs substantially better on the NEWS data, and the models produced for this thesis do not have that difference, scoring 0.0558 in contrast to the 0.0997 for the model trained on all the data. The results do show that the worst performing score for the original model was on the WIKIPEDIA data, and this is the same as the results with the newly created models as well.

Model Name	WIKIPEDIA	WIKINEWS	NEWS	ALL DATA
News	0.1028	0.0974	0.1037	0.1018
WikiNews	0.1027	0.0954	0.1008	0.0998
Wikipedia	0.1031	0.0964	0.0994	0.0993
All data	0.1015	0.1015	0.0997	0.0989
Original 2018 CAMB	0.0739	0.0674	0.0558	-

Table 4.15: Mean Absolute Error (MAE) per model with CAMB inspired features - Tested all three dev data sets individually and all dev data combined.

4.3 Results for CAMB_A

The CAMB_A system results show is the models that include additional features from the Lang-8 Learner Corpus, EFCAMDAT learner corpus and RoBERTa contextual feature information.

4.3.1 Lang-8 Features

The following results shown in Table 4.16 are the results for the model with the Lang-8 learner feature added. This is a simple frequency of if the target word or words occur in the Lang-8 learner corpus. The model was trained on all three genres with the Lang-8 frequency and then tested the combined data and the non-native and native split data.

Data Set	Precision	Recall	F-Score
All data	0.839	0.868	0.853
NATIVE	0.531	0.782	0.633
NON-NATIVE	0.477	0.738	0.580

Table 4.16: Results for classifier with Lang-8 frequency information added.

4.3.2 EFCAMDAT Features

Table 4.17 shows the performance with the five features added from the EFCAMDAT corpus. As can be seen, the non-native annotations perform worse on the NON-NATIVE data than the NATIVE data but better than the performance of the Lang-8 feature in Table 4.16. Figure 4.8 shows the confusion matrix for the added five EFCAMDAT features for all data. Figure 4.9 and Figure 4.10 show the confusion matrices for the NATIVE and NON-NATIVE data splits.

Data Set	Precision	Recall	F-Score
All data	0.860	0.855	0.858
NATIVE	0.656	0.810	0.725
NON-NATIVE	0.606	0.803	0.690

Table 4.17: Results for EFCAMDAT for all five CEFR levels as features.

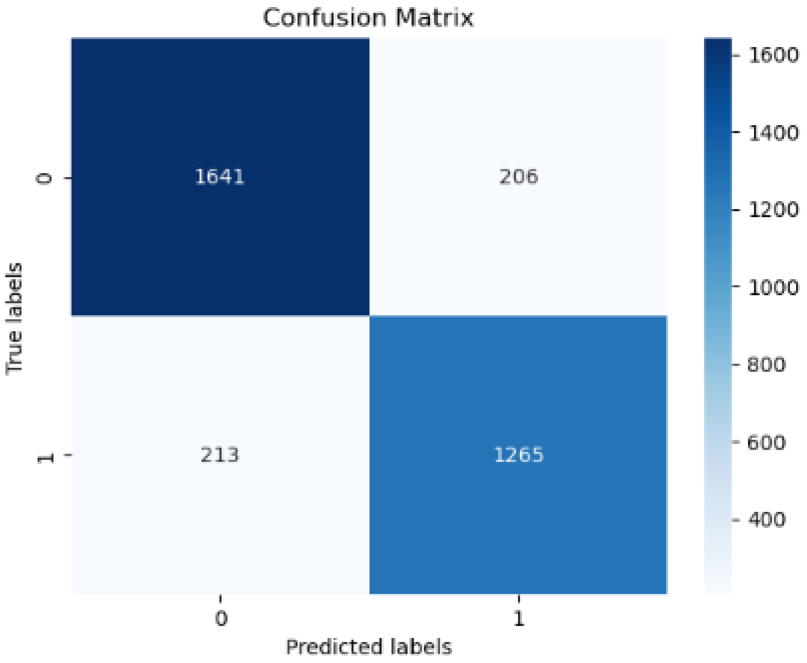


Figure 4.8: Confusion Matrix for CAMB features plus five EFCAMDAT features for all data.

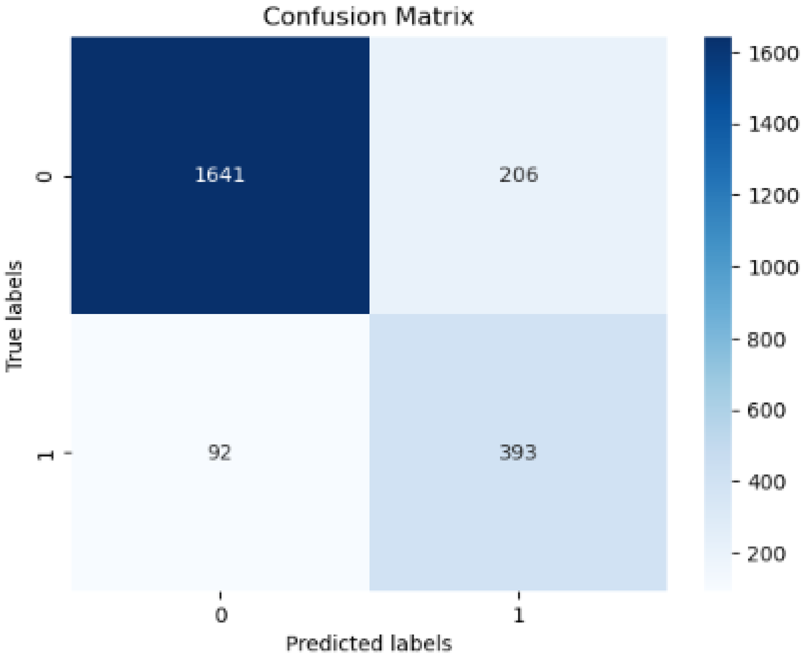


Figure 4.9: Confusion Matrix for CAMB features plus five EFCAMDAT features tested on NATIVE data.

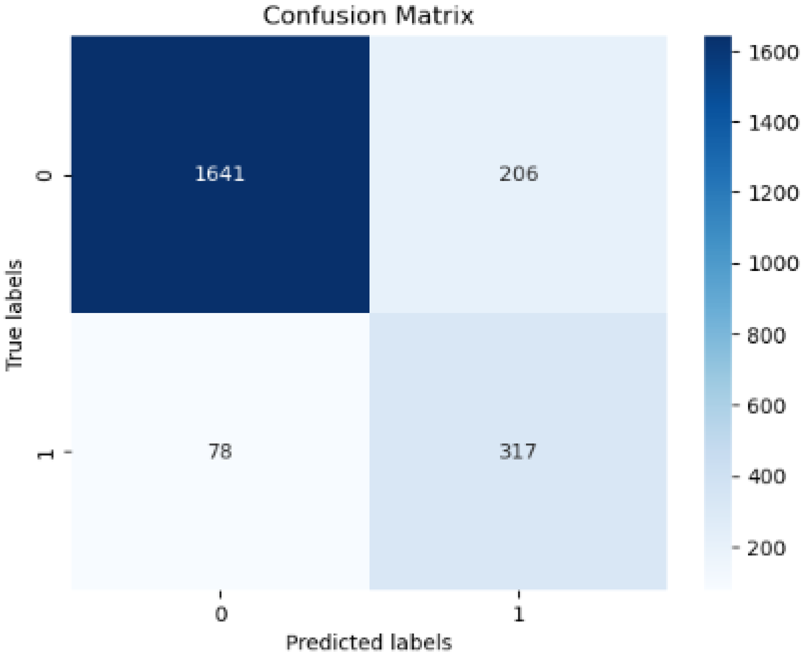


Figure 4.10: Confusion Matrix for CAMB features plus five EFCAMDAT features tested on NON-NATIVE data.

4.3.3 Results for model with RoBERTa feature added

Results with the RoBERTa embedding information added to the CAMB features are shown in Table 4.18. Total features for this model were: word length, tag,dep num, hypernyms, hyponyms, synonyms, syllables, Ogden, simple wiki, Google frequency, Subimdb, aoa, conc, fam), IMG, NPHN, Embedding and Lexicon score. The Figures 4.11, 4.12 and 4.13 show the corresponding confusion matrices. From the error analysis of these, it can be seen that all three model had the same number for predicting that the word was not complex correctly, which was 1605. Additionally, it can be seen in Figure 4.13 that when all the data was combined, the correctly predicted complex phrase went up to 750 from the very low scoring 110 and 134 for the NON-NATIVE and NATIVE data respectively. These results are surprising and possibly point to some issues with the data division.

Data Set	Precision	Recall	F-Score
All data	0.756	0.725	0.740
NATIVE	0.356	0.565	0.437
NON-NATIVE	0.313	0.516	0.389

Table 4.18: Results with RoBERTa contextual features added.

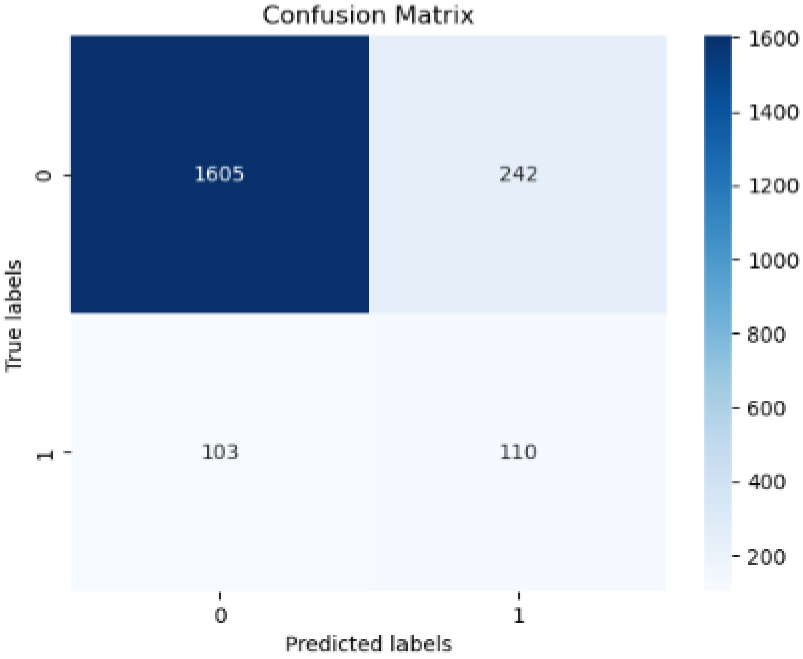


Figure 4.11: Confusion Matrix for RoBERTa features tested on NON-NATIVE data.

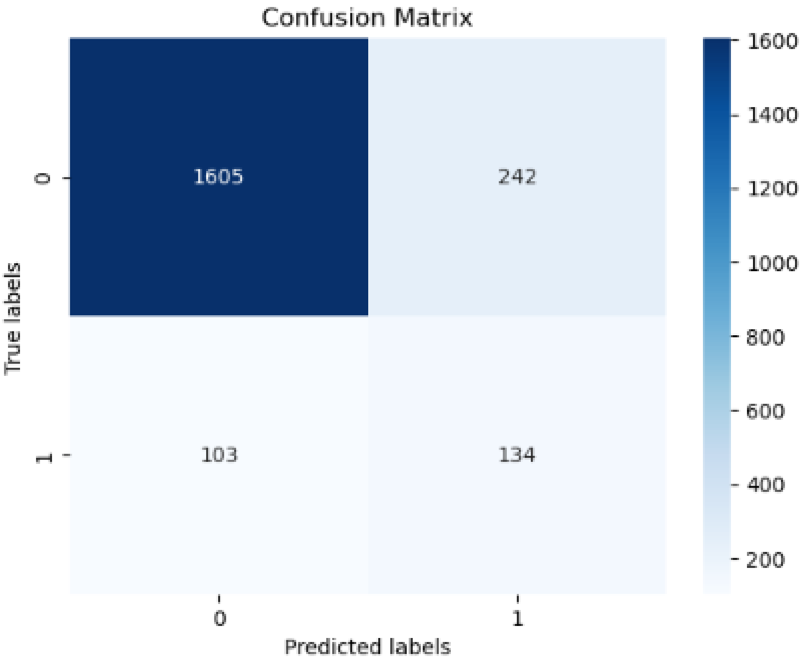


Figure 4.12: Confusion Matrix for RoBERTa features tested on tested on NATIVE data.

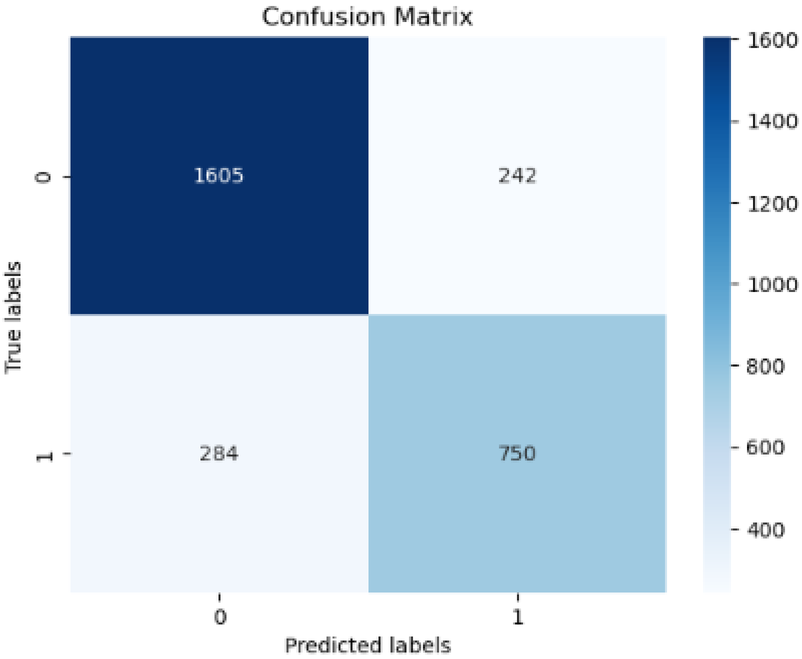


Figure 4.13: Confusion Matrix for RoBERTa features tested on tested on ALL data.

4.4 Results for the Final Model

The following results are all using the test data that was used in the original CWI-2018 completion. Table 4.19 below shows the results for the final system tested on all genre test data combined and the results for all test data split into NATIVE and NON-NATIVE. The confusion matrix for the combined test data is shown in Figure 4.14.

Data Set	Precision	Recall	F-Score
All data	0.819	0.763	0.790
NATIVE	0.440	0.603	0.509
NON-NATIVE	0.425	0.550	0.480

Table 4.19: Results for final system.

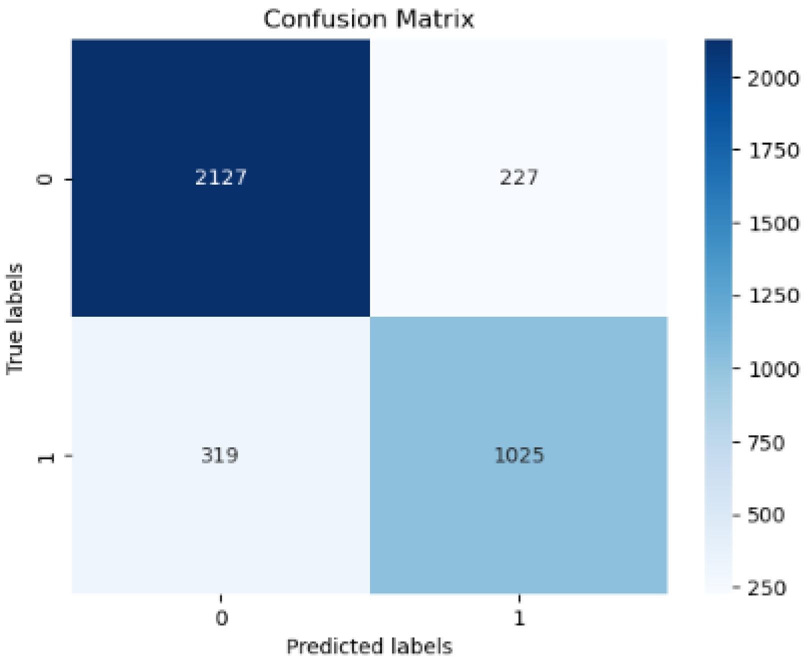


Figure 4.14: Confusion Matrix for final system tested on ALL genres test data.

4.4.1 Probabilistic results

As can be seen in Table 4.20, the WikiNews Final model and the All data Final models have the worst performance on the WIKIPEDIA data set, as both models have a score of 0.1061, which is higher than the scores of the other models. The News Final mode and Wikipedia Final models have slightly better, but the best probabilistic performance is achieved by the All data final model tested on the NEWS test data.

Model Name	WIKIPEDIA	WIKINEWS	NEWS	ALL DATA
News Final	0.1079	0.1008	0.1014	0.1025
WikiNews Final	0.1061	0.0973	0.0981	0.0995
Wikipedia Final	0.1070	0.0988	0.0980	0.1001
All data Final	0.1061	0.0982	0.0981	0.0997

Table 4.20: Mean Absolute Error (MAE) per model for the final systems.

Chapter 5

Discussion

For the baseline models, it was expected that models trained on all the data would perform best when tested on each data set as they had a larger amount of data to train on, and the genres all contained similar types of text. However, this was not the case, with the baseline models only performing best when tested on the News data. When tested with the combined development data from all three genres, the WikiNews data-trained classifier outperformed the baseline model with an F-1 score of 0.640 compared to the Baseline model trained on all the data, which scored an F1-score of 0.669. None of the simple baseline classifiers performed the best on the data that they were trained on. The baseline models may have done worse than the one used in the 2018 task due to the method of extracting Google Frequency information. In the original baseline model, this was achieved by using Google 5T. However, for the baseline model in this paper, Google Ngrams was used. Word frequency in Google 5T was one of the most important features. The Google Ngram frequency data used for the baseline models in this thesis was taken from books rather than the web. As the CWI-2018 task data is web data, this likely negatively affected performance. Further, the "search for hyphenated phrases" feature in Ngram was never made functional by the programmers. Instead, Ngram was instructed to transform requests for plots of hyphenated words into requests for plots of advanced Ngram comparisons of the component words or phrases within the original hyphenated word or phrase. For example, the phrase *"bore unusual vertebrae"* from the Wikipedia development data produces zero in Google Ngram, but the phrase *"bore a resemblance"* scored high. This would be labelled as complex in the CAMB system as the 3gram contains two words labelled complex. However, this phrase was not labelled complex by the annotators in the data by the non-native or native annotation. Multiple word phrases in the data are not annotated logically and have apparent contradictions that are not easy to make allowance for when making a human rule-based system.

Models trained on a particular genre did not perform best when tested on the same genre as was expected. Furthermore, it would be logical to suspect that data trained on all sets would be the highest performing when tested on all data. In short, for this data, the models trained on specific genres did not consistently perform better on the genre that it was trained on. This could be because the genres are not extremely different, and with the size of the data, any genre differences between models were not distinct enough. However, the poorer results on the Wikipedia data were congruent with the original CAMB system, which also performed worse in this genre. Features in the original model were changed for the Wikipedia genre data, and this data set did perform worse with the CAMB-inspired re-implementation. The model trained and

Tested on the News data scored higher than the model tested and trained on all data, with scores of 0.875 Precision, 0.866 Recall and 0.870 F-score, compared to the model trained on the combined genre training data, which scored 0.837 Precision, 0.816 Recall and 0.827 F-score. The News data trained model could have performed worse on the News data because of the way hyphenated words are treated. And this is the data set with the highest prevalence of hyphenated words, containing 213 in the training set. When examining the Google frequency scores for hyphenated words, the API did not return a score and had the default zero score. Lastly, setting the MWEs to Greedy was used by the CAMB system, and this worked to improve the performance, but it did not improve the understanding of the task. The lack of research on how to handle MWEs was a problem, and the solution to apply the Greedy algorithm to all MWEs, although simple, does not take the research forward. The CAMB-influenced model also suffered a significant drop in performance when the data was split into non-native and native. The model performed significantly worse across both sets.

When reviewing individual errors that were consistently made by all systems, it is worth noting that the same word that is repeated in a target sentence multiple times is not always labelled as complex. For example, in the WIKIPEDIA data, the word "metres" appears four times in the same sentence but is only labelled as complex once by a single native annotator giving the complex probabilistic score of 0.05. Contradictions such as this were difficult for the models to distinguish. A study by (Finnimore et al., 2019) found that across all data sets in the 2018 task, 72% of MWEs contain at least one single word with the opposite label. Every single word instance in 25% of MWE instances has the opposite label. For example, "numerous falsifications and ballot stuffing" is not annotated as complex, despite its SWs "numerous", "numerous falsifications", "falsifications", "ballot", "ballot stuffing", and "stuffing" all being complex.

For the models that were trained only on the non-native and the native data, it was surprising that the results showed that the non-native trained model improved the performance on the native data but not on the non-native data, as would be expected. Overall, the split data performed worse in all cases. There was a consistent drop in performance for both data sets compared with models that used the original data. Across all models, the non-native data also suffered more than the native. This could be due to the low inter-annotator with non-native data. The higher inter-annotator with the native data may be responsible for the smaller fall in performance when the data was split.

The CAMB_A models consisted of models built by adding extra features from the Lang-8 Learner Corpora, EFCAMDAT Corpus and contextual features from RoBERTa. Adding the Lang-8 corpora feature did not improve the F-score from the previous model's score. The F-score for all data fell slightly to 0.853 and fell considerably for the Non-native and Native split data. The EFCAMDAT features did improve slightly on the Lang-8 features, but the pattern was the same, with these features improving the Native data more than the Non-native data. On reviewing the errors in the confusion matrices in Figures 4.8, 4.9 and 4.10, it is noticeable that the model for all models performed the same for correctly predicting words that were not complex but that the model performed much better at correctly predicting words correctly as complex with the combined data.

The best-performing system at CWI-2018 was implemented in a context-independent way; thus, the intuition to improve performance was to add these missing contextual features. This was done by adding RoBERTa embedding information that aimed to give context to the target word or phrase in the sentence presented to the annotators. The results for the RoBERTa features saw a fall for the binary classification to an F-score of 0.740 when tested on all data combined. However, the most striking result was the very low score that the non-native and native split data scored, which was 0.389 and 0.437. These results were surprising, and it is unclear why the system would perform this badly on the split data. The fact that the results were so low indicated that there was some issue with the process, but no explanation for this drop in performance could be established.

In summary, contextual features from RoBERTa and additional learner corpus word frequency information did not improve the performance with the CWI-2018 data. None of these additional features beat the performance of the previous models that achieved an F-score of 0.870.

Although the investigation of the three sets of extra features shown in the CAMB_A models was hoped to improve performance, this was not the case. The highest F-score across all data was achieved in the original CAMB-influenced model trained on the News data as shown in Table 4.10. This model scored an F-score of 0.874. Thus this model was chosen as the model to implement for the final model. However, when this model was tested with the final test data, it suffered a large drop in performance to an F-score of 0.790, as shown in Table 4.19.

For the three sets of results that were achieved for the probabilistic classification, shown in Table 4.5 for the baseline, Table 4.15 for the CAMB-inspired and Table 4.20 for the final systems, the results did not mirror the binary systems for the order of performance. Overall, the probabilistic results for the baseline systems performed only slightly worse than the baseline systems used at CWI-2018. However, the difference was for the NEWS data set, which performed much worse. The 2018 baseline scores for all the genres were quite similar. Although the NEWS data did score very slightly worse in 2018, the difference was the MAE scores for WIKIPEDIA and NEWS were 0.112 and 0.1127 compared with the All-Data baseline that scored an MAE of 0.1128 for WIKIPEDIA and 0.1418 for NEWS. The probabilistic CAMB-inspired results also performed worse than the original CAMB model. With these models, the WIKIPEDIA data performed worse for both the original and the CAMB-inspired models, and the model trained on all data scored the best with the NEWS data, which was the same as the original 2018 CAMB model. Lastly, the final probabilistic models only performed slightly worse than the CAMB-inspired model that was tested using development data. This was in contrast to the results for the binary classification, which suffered a significant drop in performance. As a further step would have been to train probabilistic models for the non-native native split data, but after poor results with the binary data, this was not done.

To sum up, it was possible to build a system based on the best-performing CAMB system from CWI-2018, using open, publicly available resources. However, it was not possible to recreate the exact performance that the original researchers achieved. The original F-score for the News, WikiNews and Wikipedia data was 0.874, 0.840 and 0.811, respectively. The best-performing models were achieved by making genre-specific classifiers, as shown in the results for the genre-specific classifiers in section 4.2.3. These results that were achieved during development led to the choices for the final model, which performed considerably worse on the test set. This could be due to differences in the test data or problems during the development stage. Some drops in performance were expected as the re-implementation of CAMB did not use the exact same features. These difference in performance is likely to be because of the differences in the Google frequency and CALD learner corpora feature data.

Chapter 6

Conclusion

First of all, during the recreation of the CWI-2018 competition and the process of creating a baseline system, it was evident that MWEs and single-word classification posed a different challenge than for creating a single-word classifier. For the baseline models, trying to create an overall universal rule-based system that would deal with single words and MWEs added many technical complications. It is clear why these tasks were separated into sub-tasks at SemEval LCP 2021. Secondly, splitting the target words and phrases into non-native and native annotations caused performance to drop significantly. The results from this thesis suggest that with the amounts of annotations that were used in 2018, the features that were successful on the combined data suffered a significant drop in performance when the data was split into native and non-native. A useful further direction could be to have annotated data that used solely non-native annotators from a particular learning background to try and increase inter-annotator agreement. The split of non-native and native data did show a noticeable difference in performance, with the non-native data performing considerably worse across all models.

Appendix A

Data Statement

This data statement is written for the purposes of considering bias with the data used, as put forward by Bender and Friedman (2018) to help alleviate issues related to exclusion and bias and lead to better precision in claims about how natural language processing research can generalize. The CWIG3G2 (Complex Word Identification Task across Three Text Genres and Two User Groups) data set used was taken from the SemEval Complex Word Identification (CWI) Shared Task 2018 site[1]. The data is publicly available, and the shared task report and the system description papers are published in the BEA Workshop 2018 proceedings (CWI-2018)(Tetreault et al., 2018). Although the data sets used at CWI-2018 were in German, Spanish and French, only the English data was used in this research. The English CWIG3G2 data was collected by Yimam et al. (2017a) and covers three text genres (NEWS, WIKINEWS, and WIKIPEDIA) annotated by both native and non-native English speakers. Information on age, gender, race/ethnicity, specific native language, socioeconomic status and background education were not available. However, in the creation of the data, annotators were asked if they were native or non-native English speakers and what their proficiency levels were (beginner, intermediate, advanced). Other than the total number of annotators used, there is not much detailed further information available.

A.1 License

The data is distributed under CC-BY 4.0 license, see `https://creativecommons.org/licenses/by/4.0/fordetails`.

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

Appendix B

Links to data sets and resources

Text simplification data sets - 500 complex words each with 50 candidate simplification
https://cs.pomona.edu/~dkauchak/simplification/

SemEval-2016 Complex Word Identification
https://alt.qcri.org/semeval2016/task11/

Complex Word Identification (CWI) Shared Task 2018
https://sites.google.com/view/cwisharedtask2018/

Lexical Complexity Prediction 2021
https://sites.google.com/view/lcpsharedtask2021/call-for-participation

Lang-8 Learner Corpus
https://sites.google.com/site/naistlang8corpora

EFCAMDAT Corpora
https://ef-lab.mmll.cam.ac.uk/EFCAMDAT.html

GitHub Repository Adam_Tucker_Masters_Thesis_CWI
https://github.com/Ad262/Adam_Tucker_Masters_Thesis_CWI.git

Bibliography

E. E. Abdallah, A. E. Abdallah, M. Bsoul, A. F. Otoom, and E. Al-Daoud. Simplified features for email authorship identification. *International Journal of Security and networks*, 8(2):72–81, 2013.

M. A. Amancio and L. Specia. An analysis of crowdsourced text simplifications. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 123–130, 2014.

E. M. Bender and B. Friedman. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604, 2018. doi: 10.1162/tacl_a_00041. URL `https://aclanthology.org/Q18-1041`.

T. Brants and A. Franz. Web 1t 5-gram ver. 1. *LDC2006T13, Linguistic Data Consortium, Philadelphia*, 2006.

L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

J. A. Carroll, G. Minnen, D. Pearce, Y. Canning, S. Devlin, and J. Tait. Simplifying text for language-impaired readers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 269–270, 1999.

P. K. Choubey and S. Pateria. 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)*, pages 1006–1010, 2016.

M. Coltheart. The mrc psycholinguistic database. *The Quarterly Journal of Experimental Psychology Section A*, 33(4):497–505, 1981.

W. Coster and D. Kauchak. Simple english wikipedia: a new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669, 2011.

M. Davies. The 385+ million word corpus of contemporary american english (1990–2008+): Design, architecture, and linguistic insights. *International journal of corpus linguistics*, 14(2):159–190, 2009.

D. De Hertog and A. Tack. Deep learning architecture for complex word identification. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 328–334, 2018.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

C. Europe. *Common European Framework of Reference for Languages: Learning, Teaching, assessment: Companion volume.* Council of Europe, 2020. ISBN 9789287186454. URL `https://books.google.nl/books?id=iockEAAAQBAJ`.

P. Finnimore, E. Fritzsch, D. King, A. Sneyd, A. Ur Rehman, F. Alva-Manchego, and A. Vlachos. 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)*, pages 970–977, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1102. URL `https://aclanthology.org/N19-1102`.

W. N. Francis and H. Kucera. Brown corpus manual. *Letters to the Editor*, 5(2):7, 1979.

J. Geertzen, T. Alexopoulou, A. Korhonen, et al. Automatic linguistic annotation of large scale l2 databases: The ef-cambridge open language database (efcamdat). In *Proceedings of the 31st Second Language Research Forum. Somerville, MA: Cascadilla Proceedings Project*, pages 240–254. Citeseer, 2013.

K. J. Gilhooly and R. H. Logie. Age-of-acquisition, imagery, concreteness, familiarity, and ambiguity measures for 1,944 words. *Behavior research methods & instrumentation*, 12(4):395–427, 1980.

S. Gooding and E. Kochmar. CAMB at CWI shared task 2018: Complex word identification with ensemble-based voting. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 184–194, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-0520. URL `https://aclanthology.org/W18-0520`.

N. Hartmann and L. B. Dos Santos. Nilc at cwi 2018: Exploring feature engineering and feature learning. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 335–340, 2018.

T. Kajiwara and M. Komachi. Complex word identification based on frequency in a learner corpus. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 195–199, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-0521. URL `https://aclanthology.org/W18-0521`.

D. Kauchak. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 1537–1546, 2013.

G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

M. Konkol. Uwb at semeval-2016 task 11: Exploring features for complex word identification. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 1038–1041, 2016.

F. Kucera and W. Francis. Wn, 1967. computational analysis of present-day american english. *Providence Brown UP*, 1967.

O. Kuru. 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)*, pages 1042–1046, 2016.

J. S. Lee and C. Y. Yeung. Personalizing lexical simplification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 224–232, 2018.

Y. Lin, J.-B. Michel, E. Aiden Lieberman, J. Orwant, W. Brockman, and S. Petrov. Syntactic annotations for the Google Books NGram corpus. In *Proceedings of the ACL 2012 System Demonstrations*, pages 169–174, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL `https://aclanthology.org/P12-3029`.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019a. URL `http://arxiv.org/abs/1907.11692`.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019b.

M. Maddela and W. Xu. A word-complexity lexicon and a neural readability ranking model for lexical simplification. *arXiv preprint arXiv:1810.05754*, 2018.

S. Malmasi, M. Dras, and M. Zampieri. Ltg at semeval-2016 task 11: Complex word identification with classifier ensembles. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 996–1000, 2016.

C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

G. V. Maverick. Computational analysis of present-day american english, 1969.

D. McCarthy and R. Navigli. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*, pages 48–53, 2007.

T. Mizumoto, Y. Hayashibe, M. Komachi, M. Nagata, and Y. Matsumoto. The effect of learner corpus size in grammatical error correction of esl writings. In *Proceedings of COLING 2012: Posters*, pages 863–872, 2012.

A. Mosquera. Alejandro mosquera at semeval-2021 task 1: Exploring sentence and word features for lexical complexity prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 554–559, 2021.

K. North, M. Zampieri, and M. Shardlow. Lexical complexity prediction: An overview. *ACM Computing Surveys*, 55(9):1–42, 2023.

C. K. Ogden and G. Halász. *Basic English*. Kegan Paul Trench Trubner, 1935.

G. Paetzold and L. Specia. Lexenstein: A framework for lexical simplification. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 85–90, 2015.

G. Paetzold and L. Specia. Collecting and exploring everyday language for predicting psycholinguistic properties of words. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1669–1679, Osaka, Japan, Dec. 2016a. The COLING 2016 Organizing Committee. URL `https://aclanthology.org/C16-1157`.

G. Paetzold and L. Specia. Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569, 2016b.

G. Paetzold and L. Specia. 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)*, pages 969–974, 2016c.

G. H. Paetzold. *Lexical simplification for non-native english speakers*. PhD thesis, University of Sheffield, 2016.

A. Paivio, J. C. Yuille, and S. A. Madigan. Concreteness, imagery, and meaningfulness values for 925 nouns. *Journal of experimental psychology*, 76(1p2):1, 1968.

G. Pallotti. A simple view of linguistic complexity. *Second Language Research*, 31(1): 117–134, 2015.

C. Pan, B. Song, S. Wang, and Z. Luo. Deepblueai at semeval-2021 task 1: Lexical complexity prediction with a deep ensemble approach. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 578–584, 2021.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

S. E. Petersen and M. Ostendorf. Text simplification for language learners: a corpus analysis. In *Workshop on speech and language technology in education*. Citeseer, 2007.

L. Rello, R. Baeza-Yates, S. Bott, and H. Saggion. Simplify or help? text simplification strategies for people with dyslexia. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, pages 1–10, 2013.

F. Ronzano, L. E. Anke, H. Saggion, et al. 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)*, pages 1011–1016, 2016.

S. Sanjay, K. Soman, et al. Amritacen at semeval-2016 task 11: Complex word identification using word embedding. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1022–1027, 2016.

M. Shardlow. 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*, pages 69–77, 2013.

M. Shardlow, M. Cooper, and M. Zampieri. 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)*, pages 57–62, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-45-0. URL `https://aclanthology.org/2020.readi-1.9`.

M. Shardlow, R. Evans, G. H. Paetzold, and M. Zampieri. SemEval-2021 task 1: Lexical complexity prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16, Online, Aug. 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.semeval-1.1. URL `https://aclanthology.org/2021.semeval-1.1`.

M. Shardlow, R. Evans, G. H. Paetzold, and M. Zampieri. Semeval-2021 task 1: Lexical complexity prediction. *arXiv preprint arXiv:2106.00473*, 2021b.

I. Shatz. Refining and modifying the efcamdat: Lessons from creating a new corpus from an existing large-scale english learner language database. *International Journal of Learner Corpus Research*, 6(2):220–236, 2020.

L. Specia. Translating from complex to simplified sentences. In *Computational Processing of the Portuguese Language: 9th International Conference, PROPOR 2010, Porto Alegre, RS, Brazil, April 27-30, 2010. Proceedings 9*, pages 30–39. Springer, 2010.

L. Specia, S. K. Jauhar, and R. Mihalcea. Semeval-2012 task 1: English 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)*, pages 347–355, 2012.

A. Stolcke. Srilm-an extensible language modeling toolkit. In *Seventh international conference on spoken language processing*, 2002.

A. Tack, T. François, A.-L. Ligozat, and C. Fairon. Modèles adaptatifs pour prédire automatiquement la compétence lexicale d'un apprenant de français langue étrangère. In *JEP-TALN-RECITAL 2016*, 2016.

J. Tetreault, J. Burstein, E. Kochmar, C. Leacock, and H. Yannakoudakis. Proceedings of the thirteenth workshop on innovative use of nlp for building educational applications. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2018.

E. L. Thorndike and I. Lorge. *The teacher's word book of 30,000 words.* Bureau of Publications, Teachers Co, 1944.

M. P. Toglia and W. F. Battig. *Handbook of semantic word norms.* Lawrence Erlbaum, 1978.

M. Wilson. Mrc psycholinguistic database: Machine-usable dictionary, version 2.00. *Behavior research methods, instruments, & computers*, 20(1):6–10, 1988.

K. Wróbel. Plujagh at semeval-2016 task 11: Simple system for complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 953–957, 2016.

T. B. Yaseen, Q. Ismail, S. Al-Omari, E. Al-Sobh, and M. Abdullah. Just-blue at semeval-2021 task 1: Predicting lexical complexity using bert and roberta pre-trained language models. In *Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021)*, pages 661–666, 2021.

S. M. Yimam, S. Štajner, M. Riedl, and C. Biemann. CWIG3G2 - complex word identification task across three text genres and two user groups. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 401–407, Taipei, Taiwan, Nov. 2017a. Asian Federation of Natural Language Processing. URL `https://aclanthology.org/I17-2068`.

S. M. Yimam, S. Stajner, M. Riedl, and C. Biemann. Multilingual and cross-lingual complex word identification. In *RANLP*, pages 813–822, 2017b.

S. M. Yimam, C. Biemann, S. Malmasi, G. H. Paetzold, L. Specia, S. Štajner, A. Tack, and M. Zampieri. A report on the complex word identification shared task 2018. *arXiv preprint arXiv:1804.09132*, 2018.

M. Zampieri, S. Malmasi, G. Paetzold, and L. Specia. Complex word identification: Challenges in data annotation and system performance. *arXiv preprint arXiv:1710.04989*, 2017.