Augmenting the CoAST System with Automated Text Simplification

R J FLYNN

2021

Augmenting the CoAST System with Automated Text Simplification

Robert Flynn

A thesis submitted by the requirements of Manchester Metropolitan University
for the degree of Master of Science (by Research)

Department of Science and Engineering

2021

Acknowledgements

I would like to thank my principal supervisor Dr. Matthew Shardlow, whose guidance and support was invaluable throughout this research project. Additionally, I would like to thank Dr. Sam Sellar, my first supervisor, for providing his pedagogical perspective to the project.

Abstract

Proper comprehension of academic texts is important for students in higher education. The CoAST platform is a virtual learning environment that endeavours to improve reading comprehension by augmenting theoretically, and lexically, complex texts with helpful annotations provided by a teacher. This thesis extends the CoAST system, and introduces machine learning models that assist the teacher with identifying complex terminology, and writing annotations, by providing relevant definitions for a given word or phrase.

A deep learning model is implemented to retrieve definitions for words, or phrases of a arbitrary length. This model surpasses previous work on the task of definition modelling, when evaluated on various automated benchmarks.

We investigate the task of complex word identification, producing two convolutional based models that predict the complexity of words and two-word phrases in a context dependent manner. These models were submitted as part of the Lexical Complexity Prediction 2021 shared task, and showed results in a comparable range to that of other submissions.

Both of these models are integrated into the CoAST system and evaluated through an online study. When selecting complex words from a document, the teacher's selections, shared a sizeable overlap with the systems predictions. Results suggest that the technologies introduced in this work would benefit students, and teachers, using the CoAST system.

Contents

List of Figures

List of Tables

List of Equations

1 Introduction

Proper comprehension of academic texts is essential for students in higher education. However, many students in the UK show low-levels of literacy, with around 20% of graduates displaying literacy capabilities at, or below, level 2 (based on data from 2012) (Kuczera et al., 2016). These students will likely be unfamiliar with much of the complex language used in academic discourse. CoAST (Collaborative Augmentation and Simplification of Text) (Shardlow et al., 2021b) is a Virtual Learning Environment (VLE) that aims to help bridge this linguistic gap through a combination of the teacher's pedagogical knowledge, and machine learning tools. Teacher's are able to upload key texts to the CoAST system and provide explanatory annotations for various terms in a document, which can then be viewed by their students.

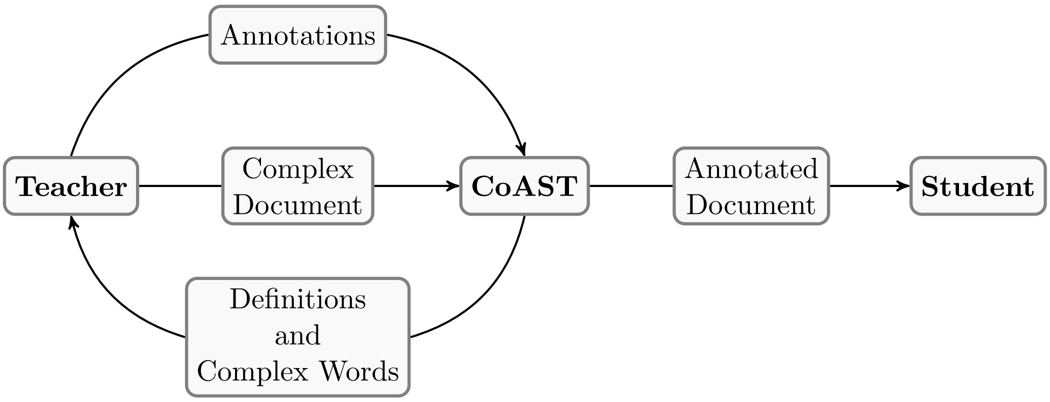
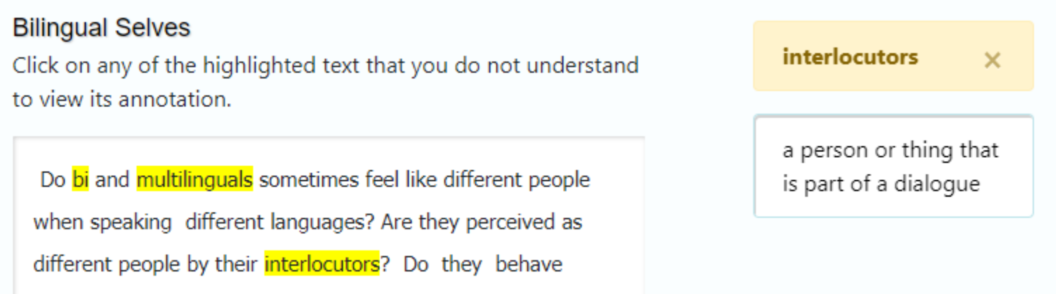


Figure 1: How CoAST mediates the student-content-teacher interaction

This sort of platform is useful for any student whose vocabulary is acting as a bottleneck to their comprehension of academic literature. English Foreign Language (EFL) learners, in particular, are a group that would benefit from a system such as CoAST. Naturally, these students will often have difficulty reading and comprehending English text. Issues understanding the meaning of certain words, and new unfamiliar words, were reasons given by the majority (32% and 29% respectively) of participants, in a study with university level EFL learners, when asked about the source of their reading difficulties (Qrqez and Ab Rashid, 2017). These difficulties may dissuade students from developing a reading habit, and effect their engagement with the literature, limiting further language acquisition (Feng et al., 2013; Qrqez and Ab Rashid, 2017). Providing additional assistance with the reading process can help break this cycle. Although this can be done in the classroom, students are able to access web-based VLEs when they choose, and come back to the text and its explanations to continually consolidate their understanding.



Bilingual Selves

Click on any of the highlighted text that you do not understand to view its annotation.

Do bi and multilinguals sometimes feel like different people when speaking different languages? Are they perceived as different people by their interlocutors? Do they behave

interlocutors

a person or thing that is part of a dialogue

Figure 2: Example of an annotated document in the CoAST system

Results from prior work involving CoAST showed the system enhanced the student-teacher-content interrelationship, improving the readers comprehension of difficult texts (Shardlow et al., 2021b). While such a system is beneficial for students, its effectiveness is dependent on the annotations that are provided by the teacher. Ensuring these annotations are comprehensive, and available for a range of texts, would be challenging for teachers, who will also have to prepare material for in-classroom teaching. Making CoAST less difficulty to properly utilize is important, as ease of use is one of the key factors in the adoption of VLEs (Babi, 2012; Davis, 1985).

This project aims to automate aspects of the annotation process in CoAST. The added assistance reduces the difficulty of locating, and providing explanations for complex words, and consequently enables teachers to annotate documents at a faster rate. This is achieved through the integration of machine learning models. These models are responsible for identifying complex text within the document, and providing the teacher with applicable definitions when writing an annotation for a complex word or phrase.

Often when students encounter challenging text they use Google to try and obtain an appropriate definition (Shardlow et al., 2021b). This relies on the student finding the correct definition that applies in the given context, and retaining it in memory. This can detract from the reading process, and if the text is particularly difficult for the student it may not be feasible. Furthermore, explanations for phrases and domain-specific terms may be hard to find or interpreted incorrectly. CoAST augments the student's reading experience with tailored explanations to simplify the reading process, a diagram depicting this interaction is given in figure 1.

This thesis begins with a review of the relevant literature. A section describing the technologies in place in the original CoAST system has been included, to convey the relevant parts of CoAST that were already functional prior to this project. This work makes three individual contributions, and the methodology and results are described separately for each of them. These contributions are as follows:

- A system used for the identification of complex words and two-word phrases. This involves two neural networks with a convolutional based architecture. These models use various features to make context-dependant predictions on the complexity of a word or phrase. CoAST previously featured a rudimentary word frequency based system, for identifying potentially complex words. This contribution looks at creating a more accurate model that is able to deal with phrases and out-of-vocabulary words.
- A retrieval system for selecting definitions for words or phrases in a given context. This model does not rely on a pre-defined sense inventory and instead attempts to select the most appropriate entry from an index of 79,030 definitions. Consequently, the model is able to select definitions for words or phrases that are not present in a typical dictionary.
- The integration of the two previously mentioned contributions into CoAST. Various features have been implemented to enable these models to function in a seamless manner. A online study is used to evaluate the effectiveness of the models at aiding CoAST's goals as a VLE, of improving reading comprehension, and language acquisition.

Finally, sections 7 and 8 look at the body of work as a whole, and discuss potential areas for improvement.

2 Literature Review

The research in this review is centred around addressing the following requirements:

1. Identify text within a document that a student may find difficult to understand
2. Provide the teacher with explanatory suggestions for complex text that would aid the understanding students understanding
3. Understand the needs of the CoAST platform and its users

The literature covered involves multiple distinct areas of research spanning the fields of technology and education. As such, each area has been reviewed and presented independently.

We look at the task of Complex Word Identification, in accordance with requirement 1, and examine key features and techniques used in this task. Literature on the tasks of Lexical Simplification and Definition Modelling is subsequently explored. These tasks both present methods that could be applied to improve a readers understanding of a text. Finally, we review work on Virtual Learning Environments and reading comprehension. These two topics are central to CoAST and help examine the requirements of a online platform that aims to improve reading comprehension, why such a system is necessary, and what proper comprehension involves.

2.1 Complex Word Identification

Complex Word Identification (CWI) is the task of identifying words that the reader may find difficult to understand. Accurately identifying complex words is crucial for many downstream simplification tasks (Shardlow, 2014). Simplifying words that the target audience finds easy to understand will often make the text more difficult or less meaningful (Paetzold and Specia, 2017; Carroll et al., 1998).

A common feature used for CWI is the frequency that a given word appears in language. Words that are used at a higher frequency are more likely to be recognised and understood by a reader (Carroll et al., 1998; Rayner and Duffy, 1986). Many corpora have been used to measure word frequencies with varying accuracy at estimating word complexity. Corpora that are representative of the everyday language use of the target demographic lead to word frequencies that are better predictors of complexity, and other psycholinguistic features such as age-of-acquisition (AoA) (Paetzold and Specia, 2016).

AoA refers to the age at which a lexical item is typically acquired (Hernandez and li, 2007) and is a feature that shows a correlation to word complexity (Quijada and Medero, 2016). Many estimates of word frequency are based on materials aimed at adult readers and therefore do not match the cumulative frequency with which readers have been exposed to words (Kuperman et al., 2012); therefore, using AoA as a feature alongside word frequency values may lead to better complexity estimations.

Word length in terms of number of characters, although not a reliable feature on its own, is often used with other features to estimate word complexity. A word's length correlates to its perceived complexity (Shardlow, 2013). A possible reason for this relationship may be due to a negative correlation between a word's length and its frequency (Bentz and Ferrer-i Cancho, 2016) in line with Zipf's law of abbreviation (Zipf, 1935).

The character sequences in complex words differ from those seen in simpler words (Popović, 2018). To investigate this relationship Popović used a mulitnomial Naive Bayes Classifier to estimate which n-gram combinations were the best predictors for word complexity. These experiments found that the combination of 2-grams and 4-grams yielded the highest overall accuracy, especially when less training data was available. 5-grams were the most accurate for the classification of simple words. The 5-gram configuration may be capturing the entire root of many simple words due to their shorter length. Incorporating word structure into CWI should aid prediction accuracy. Additionally, working with a system that does not function under the notion of probabilistic independence between n-grams would be more appropriate (Popović, 2018).

Using multiple features together can provide better complexity predictions than any singular feature individually. The CAMB system (Gooding and Kochmar, 2018) aggregated 20 different features, including many of those mentioned previously, with an AdaBoost classifier. Additional features include word n-grams represented as matrix of token counts, part of speech tags and psycholinguistic features such as concreteness and imageability taken from the MRC database (Wilson, 1988). CAMB was the best performing system at the CWI Shared Task 2018 (Yimam et al., 2018) with an average F_1 score of 0.8417 indicating the effectiveness of this approach. Experiments found that n-grams contributed the most individually to the classification framework with the highest Gini coefficient.

For the classification of phrases the CAMB system opted for a **greedy** approach, labelling all phrases as complex. This baseline outperformed labelling the phrase as complex if a number of words were found to be complex by their word-based classifier, and a n-gram based classifier. Phrases are notoriously difficult to represent due to the shifts in meaning that can take place when multiple words are used as an item (Shwartz and Dagan, 2019; Sag et al., 2002). These sorts of phrases, which can not be interprated through a composition of their constituent words, are referred to as Multi-Word Expressions (MWEs) (Rayson et al., 2010). From manually annotated datasets such as CWIG3G2 (Yimam et al., 2017) we can see that many phrases show a complexity that cannot be directly derived from their component words. As many different types of word formations can be types of MWEs they can not all be treated in the same way. Categorizing these phrases based on their linguistic patterns can allow models to more accurately estimate the complexity of MWEs (Gooding et al., 2020).

Recently, deep neural networks have surpassed many of the previous manual feature engineering based approaches to CWI (Gooding and Kochmar, 2019a; Pan et al., 2021; Bani Yaseen et al., 2021). These networks consist of multiple layers of neurons that are tuned to produce representations that allow predictions to be made about the task at hand. Convolutional Neural Networks (CNN), are a type of neural network, which were originally used on computer vision tasks (Lecun et al., 1998) with a 2D convolution matrix, but have since proved effective in other domains (Conneau et al., 2016; Abdel-Hamid et al., 2014).

Using just word embeddings as inputs, a 1D CNN approach to CWI ranked between 7^{th} and 12^{th} (depending on the corpus) in the English set of the 2018 CWI Shared Task, with an average F_1 score of 0.817 (Aroyehun et al., 2018). Including additional morphological and linguistic features can further boost the performance of these models (Sheang, 2019). Despite promising results, these approaches have used averaged features from the left and right hand sides of the target word, limiting their ability to properly assess the context. This leads to a loss of potentially important information, as averages don't account for specific distances or lexical-semantic interrelations between the target word and other text in the sentence (Flynn and Shardlow, 2021).

Transformer based language models, use a process known as self-attention to learn weightings that attribute importance to different portions of the input sequence allowing long-term token level dependencies to be captured (Vaswani et al., 2017). Fine-tuning transformer models such as BERT (Devlin et al., 2019), which are trained on a large set of corpora, has lead to state-of-the-art results in many tasks including CWI. The best performing systems in the Lexical Complexity Prediction 2021 Shared task (Shardlow et al., 2021a), made use of an ensemble of transformer based language models to produce complexity estimations for words and phrases (Pan et al., 2021; Bani Yaseen et al., 2021).

Various features and techniques that are used for the task of CWI, have been explored in this section. The research highlighted the importance of using word frequency values taken from an appropriate corpus. The SUBTLEX-UK corpus (van Heuven et al., 2014), seemed appropriate for this purpose, as it consists of subtitles taken from British television programs, making it more representative of everyday colloquial speech. AoA is another feature that is particularly useful for CWI, especially when used alongside frequency values, as it may allow models to better approximate the cumulative frequency with which the reader has been exposed to words.

Work by Aroyehun et al. (2018) and Sheang (2019), suggested that a convolutional based network would be a suitable approach for the task of CWI. The CNN using word embeddings as features saw reasonable performance, which improved with the addition morphological and linguistic features by Sheang (2019). This work still lacked many of the features discussed previously, notably AoA, character-level sequence information, and frequency values taken from a corpus consisting of everyday language.

2.2 Lexical Simplification

Lexical Simplification (LS) is a sub-task of text simplification, the goal of these tasks is to make text more accessible. LS focuses on performing simplification at a word-level, where complex words are identified and replaced with simpler alternatives (Paetzold and Specia, 2015). Much of the seminal work (Carroll et al., 1998) on this task utilized hand-curated lexical resources, such as WordNet (Miller, 1995), to obtain a list of synonyms for a given word. Word sense disambiguation (WSD), is a task that focuses on identifying the sense in which a word is used (Pal and Saha, 2015), and is used within many LS pipelines to filter out any candidates that don't match the sense of the target word. Once a list of applicable candidates has been obtained they must then be ranked by their simplicity, and suitability to the context. This stage employs many of the techniques used in CWI.

One way of generating candidate substitutions is through the use of word embeddings. These are vector representations of words based either on their co-occurrence with other words or the context in which they occur (Lavelli et al., 2004). Due to their distribution these vectors can be used to find similar words based on cosine similarity.

The REC-LS system (Gooding and Kochmar, 2019b), produced substitution candidates using word embeddings, alongside the lexical resources WordNet and The Big Huge Thesaurus[1]. Contextual embeddings from the ELMo language model (Peters et al., 2018), are then used to perform WSD, filtering out candidates that are not semantically similar in the given context.

The transformer model BERT (Devlin et al., 2019) is pre-trained with the objective of predicting masked out words within in a sequence. The LSBERT system (Qiang et al., 2020) takes advantage of this training task, for the producing suitable candidate substitutions. By masking the target complex word, and concatenating the an un-masked version of the sentence, LSBERT is able to produce a probability distribution of the masked word that considers both the target word and its context. This system achieved state-of-the-art results on LS tasks, Qiang et al. (2020) highlighted the possibility to improve the model through fine-tuning Bert on a simple English corpus.

LS can be an effective method of improving reading comprehension for second language learners, or people with neurological conditions such as dyslexia, and other readers with low levels of literacy (Rets and Rogaten, 2021; Rello et al., 2013b,a; Watanabe et al., 2009).

2.3 Definition Modelling

The term Definitions Modelling (DM) was formally introduced by Noraset et al. (2016) as “the task of generating a definition for a given word and its embedding”. Seminal approaches to this task investigated the use of definitions, which can be seen as a direct expression of word meaning, as a proxy to evaluate the validity of the semantic distribution displayed in word embeddings (Dinu and Baroni, 2014; Noraset et al., 2016). Noraset et al. approached DM as a word-to-sequence task, using a recurrent neural network architecture to predict the probability of a definition, given a word embedding. This method also employed the use of a character level CNN, to capture how certain affixes modify the meaning of the root word, and hypernym embeddings which were used as an additional input vector.

Using individual static word embeddings to generate definitions fails to account for the polysemous nature of many words or phrases which require additional context to

[1] https://words.bighugelabs.com/

infer their meaning. Additional studies attempted to address this by disambiguating the sense of the target using its surrounding context before generating a definition (Gadetsky et al., 2018).

More recently language models such as BERT (Devlin et al., 2019) have been used to produce dynamic embeddings that capture the context of words. Chang and Chen (2019) reformed DM as a ranking task, learning to map word representations to the vector space of definitions encoded via the Universal Sentence Encoder (USE) (Cer et al., 2018). This method primarily used static fastText word embeddings (Bojanowski et al., 2016) with the addition of contextualised BERT embeddings to allow the model to disambiguate between word senses. The main objective of this work was to investigate the interpretability of contextualised embeddings and the extent to which they capture word sense information. The training and evaluation of this model was done on the Oxford dictionary dataset, which contains a substantially larger amount of examples than previous datasets used for DM.

Although Chang and Chen's approach showed sizeable improvements compared to previous work on DM, their model still struggled to generalise to words that were not present in the training data. A possible reason for this lack of generalisation may be due to an inaccurate representation and distribution of definitions in vector space. Only a subset of words present in a definition are relevant to the word being defined (Noraset et al., 2016), with many unrelated definitions sharing similar phraseology such as "In relation to...". Consequently, encoders such as the USE, which are trained for semantic similarity type tasks, may be sub-optimal for DM. This approach is also not able to deal with phrases, due to the lack of a fastText embedding, and can only utilize the first three tokens of a BERT embedding.

By treating a sentence containing the target word as input and a definition as the desired output, Bevilacqua et al. (2020) utilized the pre-trained sequence-to-sequence language model BART (Lewis et al., 2019) for the task of DM and WSD. This approach employed an additional token to denote the target span allowing their model to work with phrases of an arbitrary length. Through the fine-tuning of BART, their model is able to take advantage of the knowledge learnt during pre-training to produce definitions for words or phrases not present in the training data, or in a typical dictionary. This system is evaluated on the same Oxford dictionary corpus as Chang and Chen (2019) showing some improvements when working with unseen words, on the automated benchmarks.

The assessment of natural language generation models through the use of automated metrics has been shown to be unreliable in contrast to human judgements (Novikova et al., 2017; Sellam et al., 2020). Many of these metrics also become increasingly inconsistent for sentences, such as definitions, that are often short and vary in length (Sun et al., 2019). Furthermore, as there are many correct but lexically dissimilar ways of defining a word, differences in scores between high quality systems may reflect biases or flaws in the metric, rather than true performance (Gehrmann et al., 2021). These factors make it difficult to quantitatively assess the performance of generation based approaches to DM on unseen words, where memorization of the gold definition isn't possible (Bevilacqua et al., 2020).

It is likely that (Bevilacqua et al., 2020)'s system is more adept than the automated benchmarks suggest. With their human evaluation showing much more impressive results, able to produce definitions that were superior to gold definitions in the SamplEval corpus 51.3% of the time. One of the main focuses of this work is the application of DM to discriminate between word senses, with the model achieving state-of-the-art results on this task. Unlike this system, most other WSD approaches are limited to a pre-defined sense inventory, which doesn't reflect how humans perform this task, nor the non-discrete nature of semantics (Bevilacqua et al., 2020).

Much of the mentioned literature focuses on DM as a proxy for the assessment and interpretability of linguistic representations. With the recent increase in proficiency on this task, the use of DM as a tool for improving reading comprehension has become plausible. With models that are able to tailor definitions to a specific context and produce explanations for phrases, there is the clear application of this task as a method of reading assistance. However, the potential for misinformation is a pertinent obstacle in the unsupervised deployment of such a system. When these generative models possess insufficient information to correctly define a word, they are forced to extrapolate its meaning from the context, sometimes leading to deceptive errors, a phenomenon referred to as hallucinations (Xiao and Wang, 2021; Bevilacqua et al., 2020).

Approaching DM as a ranking, or retrieval based task, would likely reduce the likelihood of deceptive errors in the models output. Natural language generation models have many more degrees of freedom than retrieval systems, which are limited to the sentences in their index. Recently, the use of dense neural networks have seen success in the field of information retrieval, particularly with the task of open-domain question answering (QA). The Dense Passage Retriever (DPR) (Karpukhin et al., 2020) utilized two BERT-based encoders, to encode queries and passages in the same low-dimensional and continuous space, allowing for efficient retrieval through a maximum inner product search. This method saw state-of-the-art results on the task of open-domain QA, and DM could essentially be viewed as a question answering task.

2.4 Virtual Learning Environments

Virtual Learning Environments (VLEs) have been defined as online applications that enable various kinds of interaction between students and their tutors (JISC, 2002). The quality and usability of VLEs are key factors affecting the student and teacher's satisfaction when using these products (Babi, 2012). Similar findings are reported from the Technology Acceptance Model (TAM) (Davis, 1985) in which perceived usefulness and perceived ease of use are found to be major determinants for the actual use of a particular system.

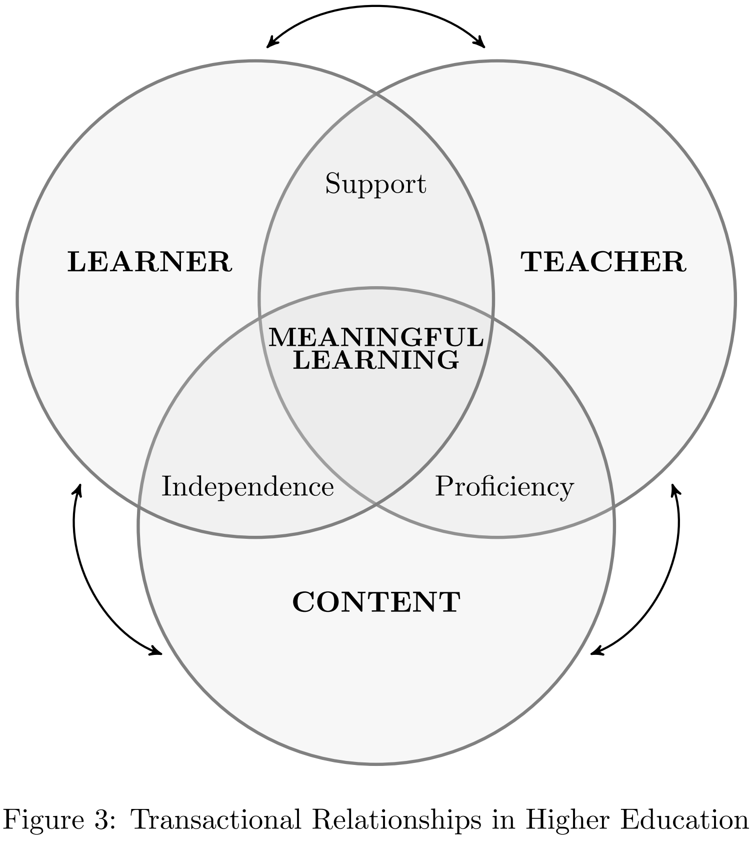


Figure 3: Transactional Relationships in Higher Education

The reliance on VLEs in education has increased dramatically over the years, particularly with the move to blended and distance learning during the COVID-19 pandemic. Education is formed from the transactional relationship that takes place between between students, teachers and content (see figure 3) (Anderson and Garrison, 1998). Each of the different elements of this interaction have certain advantages and disadvantages. Humans are able to incite enthusiasm and engagement in a subject, while using their content expertise to diagnose and rectify issues the learner is having; content is readily available 24-hours a day, making learning more accessible, and allowing the learner to return to the same consistent information and repeat a task/activity multiple times (Stein, 2014). VLEs, if implemented properly have the ability to enhance the student-content-teacher interrelationship. The use of interactive systems can make content more engaging, and integrating the teachers pedagogical knowledge with existing content in response to system and user feedback can help personalize the learner's experience.

VLEs are able to provide a multimodal environment not available through the medium of printed text. Utilizing these multiple formats for improving reading comprehension can increase engagement and promote understanding (Ortlieb et al., 2014). Reading in the context of education is primarily a student-content interaction, although many students may require some assistance with their understanding of a document. This will be especially true of second language learners. In a study investigating the role of VLEs in improving reading comprehension amongst English Foreign Language (EFL) learners, students in the virtual group showed the most progression. Their performance was compared against a control group and a blended learning group (Meulenbroeks, 2020). Blended learning involving both instructional support in the classroom, and assignments through VLEs, can also be beneficial for EFL learners reading comprehension (Behjat et al., 2012; Szymańska and Kaczmarek, 2011).

It is essential that VLEs are easy to use and navigate. There are often technical difficulties for both students and teachers affecting the uptake of these systems. The preparation of sufficient online content can also be difficult for teachers to manage, while also providing in-classroom teaching (McCown, 2010). Students with dyslexia may benefit from the tools that VLEs can provide, but many of these systems are often difficult to use and information dense, in terms of their content and interface (Sennett, 2016; Habib et al., 2012).

It is important that these learning environments adhere to accessibility guidelines, and provide tools to access pronunciations and definitions for various special ambiguous words, in order to benefit these students (Habib et al., 2012). With proper design and development, VLEs can be a useful tool for supporting the acquisition of language, and other learning, for people from all backgrounds.

2.5 Reading Comprehension

Comprehension, which is the main objective of the reading process, can be defined as "the act of constructing meaning with oral or written text" (Kamil et al., 2010, p. 200). Reading comprehension involves the convergence of multiples processes. Landi et al. (2013) considers this to begin with the decoding of a word into speech, and the retrieval of its meaning. The reader must then connect multiple words together to decipher the meaning conveyed by a sentence. Finally, to obtain an overall understanding of the text, multiple sentences and paragraphs must be linked together and conceptualized as a whole (Landi et al., 2013). Readers can struggle at different steps of this process, however the higher-level components of comprehension depend on an understanding of the lexical components of a piece of text (Perfetti and Adlof, 2012; Shankweiler et al., 1999). Those with cognitive disorders such as dyslexia will often have difficulty with the initial decoding step of reading, yet display normal comprehension in terms of oral language (Shankweiler et al., 1999; Landi et al., 2013).

The engagement with the literature, and a students motivation towards reading, affects their reading frequency which inevitably plays a key role in their comprehension and acquisition of language (Guthrie et al., 1999). In a study investigating reading comprehension amongst English foreign language learners 51% of students attributed their reasons for not reading to a lack of habit, with 25% reporting the difficulty of the text as their reason. For the majority of students these difficulties were attributed to ambiguous or unfamiliar words (Qrqez and Ab Rashid, 2017). It could be argued that the difficulty of many texts leads to a self perpetuating feedback loop in which complexity, and the frequency of unknown words dissuades the student from reading which prevents any further language acquisition.

If student's struggle to understand the content of a text or the literature has little personal meaning to them they may disengage (Kamil et al., 2010, p. 675), and ultimately "engaging with any text relies on realizing its potential for meaning" (Halliday and Hasan, 1976 cited in Bunch et al., 2014). In an experiment investigating the relationship between text difficulty and mind wandering, which could be seen as the opposite of engagement, Feng et al. (2013) found a significant positive correlation (B=.110, SE=.039, $p<.01$); additionally, mind wandering was only significantly impactful to reading comprehension on difficult texts (B=−.625, SE=.158, $p<.001$) (Feng et al., 2013). A similar study supported these findings while reporting that topic interest helps mediate the effect of a text's difficulty on mind wandering (Soemer and Schiefele, 2019).

According to Fillmore and Fillmore (2012) the language used in complex academic texts differs in many ways to that of colloquial speech. This poses a problem for students that are second language learners or have an inadequate level of literacy, who won't have had a chance to encounter the complex and information dense language of many academic texts. Furthermore, providing these students with simplified versions of texts will likely prevent them from obtaining a full grasp of the language used in academic discourse. It is to be expected that many of these students will require some form of instructional support in order to learn how to procure the information that is conceptualized in these texts (Fillmore and Fillmore, 2012).

3 Original System

This work is built on top of the pre-existing CoAST system (Shardlow et al., 2021b). In order to properly convey my contribution to the project this section will explain the functionality that was already in place.

3.1 Technologies Used

- **Angular:** The front-end of CoAST runs under an Angular framework. This is used for the creation of dynamic web applications with TypeScript and HTML.
- **Node.js:** The back-end runs on Node.js, a server side runtime that uses JavaScript.
- **MongoDB:** CoAST uses MongoDB as its database.

3.2 Interface and Functionality

CoAST features a log in and sign up page which allows users to register as either a student, teacher or admin. Students are able to view documents with annotations that the teachers have added. Teachers and Admins are able to add documents and perform annotations. Figure 4 depicts the interface used for adding posts to CoAST. Additionally, there is an Analytics page which allows non-student users to view details of words students have clicked on, and posts that students have viewed.

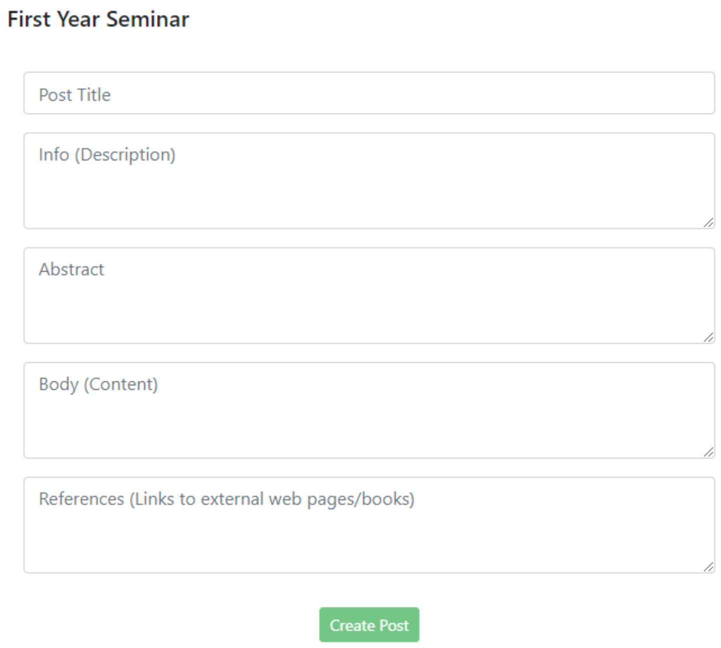
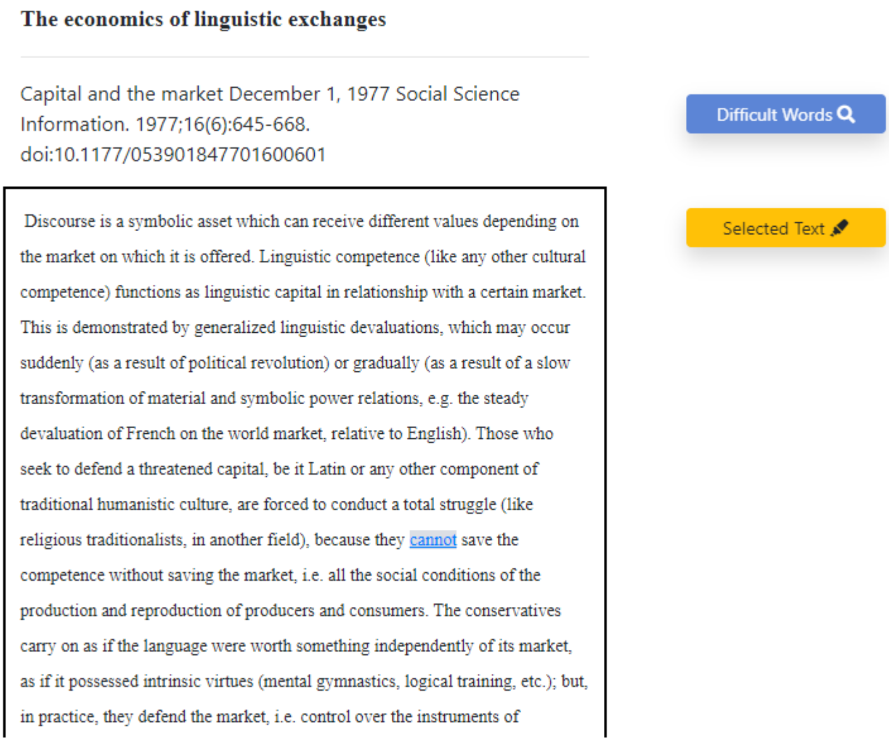


Figure 4: Interface for adding posts to CoAST

3.3 Complex Word Identification and Annotations

CoAST already had some limited CWI functionality set up. This system worked using a frequency thresholding method. Teachers are able to choose from 3 difficulty settings, for which the text is highlighted. At start-up, words and their frequency values found in text file, are loaded into different arrays based on their difficulty/frequency. When a difficulty level is selected by the teacher a request is sent to the back-end server and the document's text is parsed, word by word. If a word does not exist in the array corresponding to the selected difficulty then it is flagged as complex and sent back to the client, where it is highlighted. A consequence of this mechanism is that all words not present in the frequency list are flagged as complex. This may cause simple words to be incorrectly highlighted, leading to a seemingly inconsistent selection.



The economics of linguistic exchanges

Capital and the market December 1, 1977 Social Science Information. 1977;16(6):645-668. doi:10.1177/053901847701600601

Discourse is a symbolic asset which can receive different values depending on the market on which it is offered. Linguistic competence (like any other cultural competence) functions as linguistic capital in relationship with a certain market. This is demonstrated by generalized linguistic devaluations, which may occur suddenly (as a result of political revolution) or gradually (as a result of a slow transformation of material and symbolic power relations, e.g. the steady devaluation of French on the world market, relative to English). Those who seek to defend a threatened capital, be it Latin or any other component of traditional humanistic culture, are forced to conduct a total struggle (like religious traditionalists, in another field), because they cannot save the competence without saving the market, i.e. all the social conditions of the production and reproduction of producers and consumers. The conservatives carry on as if the language were worth something independently of its market, as if it possessed intrinsic virtues (mental gymnastics, logical training, etc.); but, in practice, they defend the market, i.e. control over the instruments of

Figure 5: Original CWI example

Figure 5 demonstrates the pitfalls of the CWI in the original version of CoAST. Here the beginner difficulty is selected, which highlights words that would be deemed complex on all difficulty levels. From the figure we can see that the only word highlighted is “cannot”. Despite this being a fairly simple word, it is a uncommon variant of the contraction “can’t”, and therefore has a low enough frequency to be featured on the beginner setting.

Many words in language can be modified through the addition of a prefix or a suffix. Words in a document may be present in the frequency list with a high frequency, but with modifications, meaning these words will be incorrectly highlighted. Furthermore, the parsing in place only splits the document into words based on spaces so words with punctuation will be flagged as complex even if they exist in our frequency file, with a high frequency, without said punctuation. When the document is received by the back-end it is converted to lowercase characters, however the same operation is not performed on words from the frequency file, which could also lead to some edge cases. Words on our front-end are also parsed to remove any punctuation or symbols, as this has not been performed on the backend these words will not be highlighted despite being flagged as complex. Because of these factors many words are missed out by the system. The majority of words flagged as complex are high frequency words in our file but used with punctuation.

As this system relies on a frequency threshold, it would be unfeasible to perform CWI on phrases, due to the number of possible combinations. The front-end does allow teachers to add their own annotations for phrases. However, the document is iterated word by word so the annotations for phrases will never be seen and these phrases will not be highlighted. This is a particularly pertinent issue for CoAST as domain-specific phrases are often difficult for students to understand. Another issue with our parsing and highlighting system is that when the difficulty level is changed by the teacher, new difficult words are added on top of the words already highlighted. If the teacher originally selects the beginner difficulty, then decides a higher difficulty is appropriate they won't be able to see any change, as the beginner words are not removed.

Teachers are able to provide annotations for words by typing into a text box on the sidebar of a document. These annotations are stored in the database and presented to the student when they log in and view the document.

3.4 Further Work

The previous subsection identified some issues that will need to be addressed as part of this project. CWI is a key component of this work, and in section 4 we will discuss the implementation of a new system that is more robust, accurate, and capable of handling phrases. Changes to CoAST to allow for any new functionality, and the resolution of current issues, will be covered in section 6.1. Additionally, Infrastructure and UI changes will need to be made in order to properly integrate the new CWI and annotation assistance technologies.

4 Complex Word Identification

The following gives an explanatory overview of the complex word identification (CWI) system used within CoAST. The code for these models has been provided on GitHub[2].

The work in this section is largely taken from a paper (Flynn and Shardlow, 2021) I have previously published as part of the Lexical Complexity Prediction (LCP) 2021 shared task (Shardlow et al., 2021a). Some passages have been reproduced verbatim from this paper (Flynn and Shardlow, 2021).

4.1 Methodology

4.1.1 Dataset

This work makes use of the CompLex dataset (Shardlow et al., 2020). Many of the previous datasets for CWI used binary annotations to denote word complexity. However, Human judgements of complexity exist on a continuous scale. CompLex provides complexity annotations for individual words and phrases on a 5-point Likert scale that is normalised between 0 and 1. Excerpts from three domains (Bible, Europarl and Biomed) are used in this dataset.

The LCP 2021 shared task (Shardlow et al., 2021a) utilized CompLex to compare the performance of various models on the complexity estimation of single words and two-word phrases. Details of the train/dev/test used for the training and evaluation of the models is provided in table 4.1.1.

Task	Train	Dev	Test
Single Words	7662 / 3478	429 / 213	917 / 429
Multi Words	1517 / 1270	99 / 76	184 / 142

Table 1: CompLex dataset for the LCP 2021 shared task. Each cell displays the values for the number of instances / unique words.

4.1.2 Features

Both model use a range of features that were selected based on previous work and through exploring their effectiveness on the validation data with a linear regression model. Details of these experiments are given in table 2. The Base features refer to: log frequencies (Subtlex), AoA, word length, syllable count and corpus type. These features as well

[2] https://github.com/robflynnyh/CNN-LCP-Shared-Task-2021/blob/main/models.py

as additional embeddings that were included in the final model are explained below:

Word Frequency values are taken from the SUBTLEX-UK database (van Heuven et al., 2014). Logarithmic Zipf frequencies were chosen for this task, based on previous work on CWI (Zampieri et al., 2016), and the Zipfian distrubution displayed in language (Zipf, 1949). Additionally, the experiments using linear regression showed significantly higher results when using logarithmic frequencies.

Age of Acquisition values, estimating the age at which a word is typically acquired (Kuperman et al., 2012; Brysbaert, 2012).

Word-Level Features measuring the length, and number of syllables of the target word (Brysbaert, 2012).

Corpus Type is included and represented to the models in the form of a one-hot embedding. Our dataset includes extracts from three different sources, which may vary in their complexity.

Pre-trained Embeddings representing word and character sequences. 50d GloVe (Pennington et al., 2014) word embeddings were chosen as initial experiments conducted with the training data showed that GloVe embeddings with higher dimensions yielded lower accuracy's. Both types of embeddings were also tested with the final model with similar results. To represent character sequences and allow inferences to be made about words sharing similar morphologies, Char2Vec [3] 50d character embeddings are also used.

Features	Pearson	MSE	R2
Frequency(Web1t)	0.3087	0.0174	0.0782
Frequency(Subtlex)	0.3206	0.0175	0.0736
Log Frequency(Subtlex)	0.6814	0.0102	0.4580
Syllables	0.3286	0.0170	0.1017
Word Length	0.1589	0.0185	0.0207
Base	0.7136	0.0094	0.5045
GloVe 300D	0.7246	0.0091	0.5189
GloVe 50D	0.7291	0.0089	0.5308

Table 2: CWI linear regression feature experiments on validation data

4.1.3 Preprocessing

Firstly all the feature sets have to be normalized within in a similar range that is close to zero. Only the distribution of a feature is useful to a model. If each feature given is on a different scale then features with a higher mean will initially be weighted higher, decreasing performance and increasing the training time. Our model will use ReLu as

[3] https://github.com/IntuitionEngineeringTeam/chars2vec

its activation function with the formula: $f(x) = \max(0, x)$. Because of the nature of this function, features with a high mean will be disproportionately represented within the hidden units of the network.

$$\frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{1}$$

To scale the features between 0 and 1, while maintaining their original distribution, the Min-Max formula (Shown in equation 1) is used on all of the linguistic features, other than the word length. In retrospect, this could also be applied to the word length feature by taking the longest length word as the max value. Instead word lengths are divided by 10, which puts them in a similar range to the other features.

Before any features are extracted, all the non-alphanumeric characters are removed from the data. This is to prevent any words which would over-wise have had an associated feature, being classed as out-of-vocabulary due to the presence of a symbol. For example, hyphenated words will likely not have an associated frequency value, however their component words might.

Both models take as inputs the features for the target word(s), and the averaged features for the left and right context of the target text. For target word/s positioned at the beginning or end of a sentence a zero vector of size 107 is constructed and used for the left or right context. For out-of-vocabulary words a zero vector is used for the word embeddings and other features are imputed using the mean values from their datasets.

For each sentence, or instance, the vectors for the target text and its context are stacked to produce a 3x107 matrix (left context — token — right context) for single words or a 4x107 matrix for the multi-word model (left context — token 1 — token 2 — right context).

4.1.4 Models

Both models are produced using the Keras library[4] (version 2.4.3). The data was processed via a batch size of 50 and both models are configured with early stopping set to 1000, and model checkpointing based on the validation loss.

Single-Word Model

This model is used to produce complexity estimations for individual words. It takes 3 feature sets as inputs in a 3×107 matrix. Features for the target word act as the second

[4] https://keras.io/

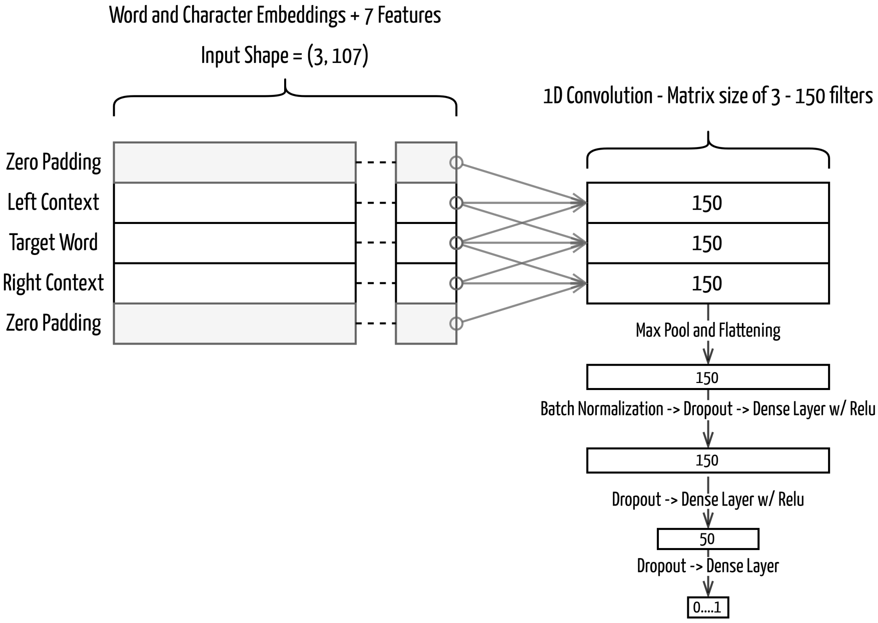

Word and Character Embeddings + 7 Features
Input Shape = (3, 107)
1D Convolution - Matrix size of 3 - 150 filters
Zero Padding
Left Context
Target Word
Right Context
Zero Padding
150
150
150
Max Pool and Flattening
150
Batch Normalization -> Dropout -> Dense Layer w/ Relu
150
Dropout -> Dense Layer w/ Relu
50
Dropout -> Dense Layer
0....1

Figure 6: CWI single word model architecture

input; averaged features for the left and right contexts of the target word are used for the first and third inputs respectively. The inputs are fed to a 1D convolutional layer which adds zero padding to each side, effectively creating a 5×107 matrix. This layer uses a kernal size of 3 with 150 output filters and ReLu as its activation function. Global max pooling and a flatten operation is applied over the output of the convolution followed by batch normalization. Three dense layers with sizes of 150 (ReLu), 50 (ReLu) and 1 (Linear) are then used with a Dropout of 0.5 applied before each dense layer. The final linear layer gives us a complexity prediction for the target word.

A diagram depicting the model's architecture is given in figure 6. Mean squared error is used as the loss function and Stochastic Gradient Descent as the optimizer, with a learning rate of 0.01 and momentum of 0.6 with Nesterov accelerated gradient enabled.

Multi-Word Model

For multi-words a second model is used to assess the complexity of two word phrases. This model is designed to build a representation of each phrase, which is then fed to the pre-trained single-word model. The dataset for single words is significantly larger than the data available for phrases. The use of both models allows the multi-word model to take advantage of the information learnt through training on single words. Figure 7 depicts the architecture of this model.

Features for the averaged left context, target word one, target word two and the averaged right context are used as inputs for this model. For each target word a 3×107

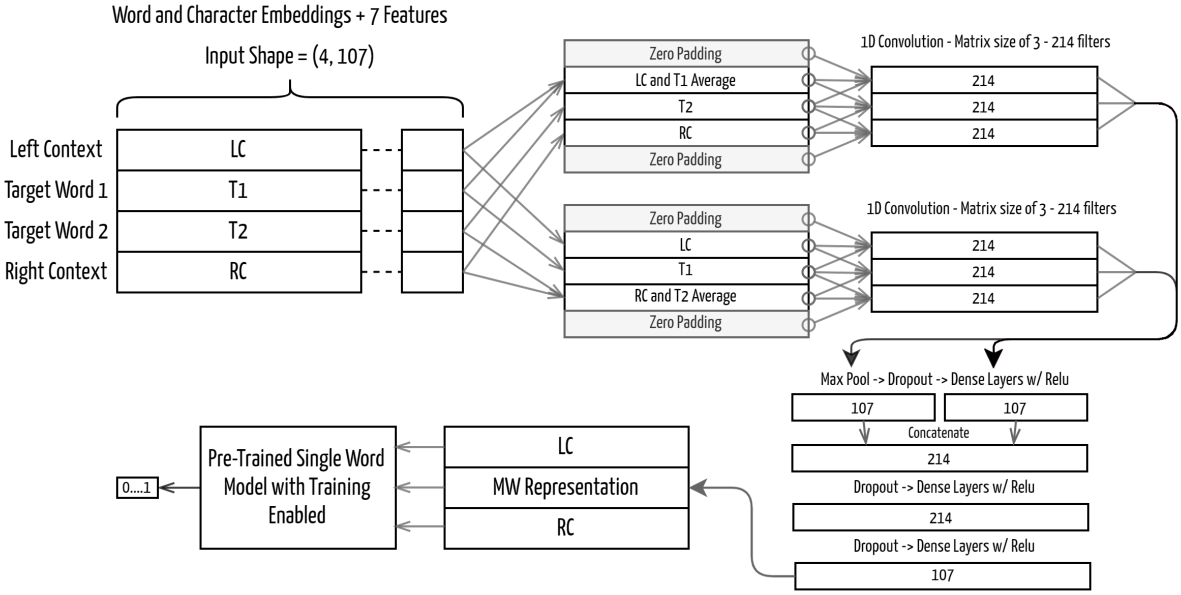

Word and Character Embeddings + 7 Features
Input Shape = (4, 107)
Left Context
Target Word 1
Target Word 2
Right Context
LC
T1
T2
RC
Zero Padding
LC and T1 Average
T2
RC
Zero Padding
1D Convolution - Matrix size of 3 - 214 filters
214
214
214
Zero Padding
LC
T1
RC and T2 Average
Zero Padding
1D Convolution - Matrix size of 3 - 214 filters
214
214
214
Max Pool -> Dropout -> Dense Layers w/ Relu
107
107
Concatenate
214
Dropout -> Dense Layers w/ Relu
214
Dropout -> Dense Layers w/ Relu
107
LC
MW Representation
RC
Pre-Trained Single Word Model with Training Enabled
0....1

Figure 7: CWI multi-word model architecture

matrix is produced by taking an average of the other target word and the left or right context, depending on its positioning. This weights the other target word higher than the rest of the context. These two matrices are then used as inputs for two 1D convolutional layers. Each layer has a filter size of 214 but is otherwise the same as the convolutional layer used in the single-word model. Global Max Pooling followed by Dropouts of 0.3, and dense layers with 107 neurons and ReLu activation are applied to the outputs of the convolutions. These outputs are then concatenated along the last axis to form a layer with a size of 214. This is then fed to two dense layers with ReLu activation and sizes of 214 and 107. Dropouts of 0.5 are used before each dense layer.

The final output of size 107 is then concatenated along the first axis with the original left and right contexts to form the input for a pre-trained single word model with training enabled. This model uses the Adam optimizer with default parameters and MSE as the loss function.

4.2 Results and Analysis

4.2.1 Single Word Model

As shown in Figure 8 the single word model struggles to accurately predict values for words of a high complexity, and also displays difficulties for words of a complexity of less than 0.1. The training and evaluation data contains less examples of very simple or complex words. This may cause the model to skew its predictions closer to the av-

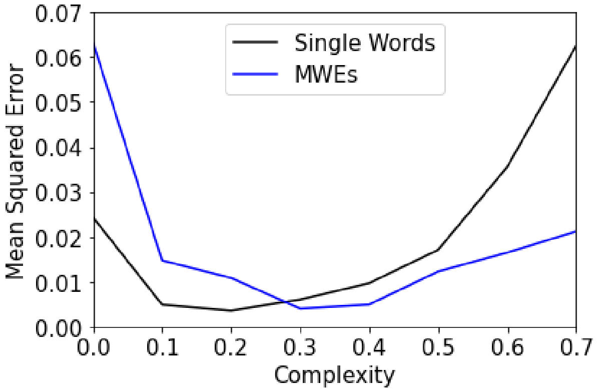


Figure 8: MSE CWI for both models across different complexities

erage displayed in the training data, if insufficient data is available to make a confident estimation. The complexity of these extremities is often highly dependant on the context, making them more challenging to assess.

Corpus	**Pearson**	**MSE**	**R2**
All	0.7389	0.0074	0.5398
Bible	0.7085	0.0085	0.4948
Biomed	0.7828	0.0087	0.6050
Europarl	0.6807	0.0055	0.4562
JUST-BLUE	0.7886	0.0062	0.6172

Table 3: CWI single word model results

Table 3 presents the results for this task on each of the domains and the task as a whole. The prediction accuracy varies significantly across the different sources. Results from the best performing team in the LCP shared task are given for comparison (Bani Yaseen et al., 2021).

As the model only uses an average of the features present in the left and right context of the target word, it is unable to differentiate between tokens that are influential to the target words complexity and ones that are not. Because of this equal weighting of words in the context, the models accuracy can be negatively affected by an abundance or lack of stop words in the sentence. Very complicated or simple words in sentence that are not related to the target word, and don't share a similar complexity can also cause the model to over- or under-predict the target word's complexity. The mechanism by which the model assesses the context may partly explain the variance in accuracy on each domain. Interestingly, the sub-analysis showed that the model shows a better correlation for those tokens without a word embedding, yielding a Pearson correlation of 0.7804 and a MSE of 0.0071. Generally these out-of-vocabulary words are more complex so the model is using the lack of a word embedding as a feature when making predictions. Although this shows a better correlation overall it could lead to false positives in specific cases where the out-of-vocabulary word is of a low complexity.

4.2.2 Multi-Word Model

Corpus	Pearson	MSE	R2
All	0.7611	0.0102	0.5770
Bible	0.7173	0.0113	0.5106
Biomed	0.7980	0.0141	0.6317
Europarl	0.5799	0.0060	0.3089
DeepBlueAI	0.8612	0.0063	0.7389

Table 4: CWI results for multi-word model

As shown in Figure 8 the multi-word model is much less accurate for very simple MWEs of a complexity less than 0.1. However, for more complex words the predictions remain fairly accurate. This model is able to asses the way in which the words in a phrase interact with each other and to some degree the rest of the sentence. This additional contextual information may increase the model's capacity to evaluate more complex words. Only 1.65 percent of phrases in the training data were of a complexity of less than or equal to 0.1 which could explain the inaccuracy in this range.

Table 4 presents the results across each of the different domains present in the dataset. Results from the best performing team in the LCP shared task are given for comparison (Pan et al., 2021). The model used for MWEs makes use of a fine-tuned instance of the single-word model. Consequentially, incorrect associations from the single-word model may have been carried over to this model. The results show a similar variance across domains to task 1, although it struggles more significantly on the Europarl sub-corpus. Compared to the other domains, Europarl's complexity values show a much smaller standard deviation than the other sub-corpora (0.093 compared to 0.196 and 0.152, on biomed and bible). The variation of complexities may play a role in the models effectiveness at making accurate predictions across the domains.

MWE Type	Pearson	MSE	R2
A-N (115)	0.7654	0.0115	0.5801
N-N (56)	0.7414	0.0091	0.5293

Table 5: CWI results for the different MWE formations. A-N: Adjective-Noun. N-N: Noun-Noun.

Table 5 presents the results across different MWE formations. The number of occurrences of each part-of-speech formation is denoted in brackets, MWE types with less than 10 occurrences were omitted from the table. The model performs marginally better on Adjective-Noun MWE formations. It is likely that the complexity of Noun-Noun phrases is often more ambiguous, as adjectives do not usually have as much impact on a phrases meaning compared to nouns.

4.2.3 Ablation Experiments

To investigate the contribution of each of the features used in the models, the single-word model has been re-trained using the same parameters, with various features removed. Table 6 shows the results of each experiment using different feature sets, the results for the entire model are also given for reference.

Features	Pearson	MSE	R2
Char2Vec	0.3465	0.0143	0.1185
AoA	0.6002	0.0104	0.3589
Freq	0.6199	0.0100	0.3829
Base Excluding Freq	0.6620	0.0091	0.4369
Freq + AoA	0.7004	0.0084	0.4832
GloVe 50D	0.7179	0.0080	0.5077
All Excluding Glove50D	0.7231	0.0079	0.5136
Glove 50D + Char2Vec + Freq	0.7250	0.0077	0.5248
Base Excluding Corpus	0.7326	0.0075	0.5326
All Excluding Char2Vec	0.7336	0.0075	0.5340
All Excluding Corpus	0.7366	0.0075	0.5383
Base	**0.7393**	**0.0074**	**0.5458**
All	0.7389	**0.0074**	0.5398

Table 6: CWI ablation experiments, analysing the effectiveness of different features

The Base features refer to: frequency, AoA, word length, syllable count and corpus type. Freq is used as an abbreviation of frequency. From these tests we can see that the model actually performs marginally better when only using the Base feature set, a surprising result. From the Base features the frequency values are the most informative with AoA falling close behind, when used together they act synergistically. The GloVe 50D embedding is the most effective *individual* feature, and reasonable results can start to be seen with the inclusion of frequency values and the character embeddings. It is clear that the corpus type also provides additional useful information, that aids prediction accuracy, with a drop in the R2 score being displayed upon its removal from the base features set.

The higher performance on the base set, is likely due to insufficient training data, or rather the amount additional information that the embeddings provide regarding our data does not justify the increased dimensionality. When the number of dimensions is increased the volume of space that the data occupies also grows, this sparsity makes it difficult to make generalisations from our data. The amount of data required for a task grows exponentially with the amount of dimensions that the data takes on, this phenomena has been coined the “curse of dimensionality” (Bellman, 1957).

When using this model within CoAST the corpus type feature will not be available. The model using all the features is not as effected that effected by the removal of the corpus type. Although the difference here is negligible, its performance is better than the base set with corpus type removed.

To further investigate the performance of the base features the Multi-Word model has been retrained, results from these models are given in table 7. Once again the base feature set actually shows the best performance on the multi-word model. Corpus type is a very sizeable contributer to performance in the base set, with the Pearson correlation dropping by 0.0575 upon its removal, suggesting each domain has a varied set of phrases that differ in complexity. Out of the feature sets without corpus type, we see a similar trend to the single-word model, with All marginally outperforming the base set.

Features	Pearson	MSE	R2
Base	**0.7854**	**0.0097**	**0.5993**
Base Excluding Corpus	0.7279	0.0116	0.5211
All	0.7611	0.0102	0.5770
All Excluding Corpus	0.7362	0.0113	0.5299

Table 7: CWI multi-word model ablation experiments

Overall for use in CoAST the results suggest that there will be relatively little difference between using all features or just the base set (both without corpus type). The data that we test on here however is relatively clean with most words existing in our frequency or AoA datasets, which are the biggest contributors to the performance with the base set. There are 400,000 Glove Embeddings for different words compared to 160,021 frequency values or 51,715 AoA values. The Char2Vec embeddings also provide data for every word as they are generated to represent the character sequence. These additional features will likely be valuable in a use cases such as CoAST where they can provide some information about unknown words or misspellings, on which to base the predictions. Although, there has not been a sub-analysis done on the performance with these two features sets on words that have missing values in our datasets.

4.3 Discussion

Overall both models produced reasonable results, in a comparable range to other approaches to this task. The effectiveness of the base feature set was a surprising finding of the ablation experiments, although this was partly dependent on the corpus type feature, which would not usually be available. The character embeddings contributed fairly little to the models performance, finding alternative ways to represent character sequences to the model would potentially be more effective.

The models may have worked better with a larger amount of data. Most CWI datasets are produced using binary annotations, which makes them difficult to directly use for this task. However, pre-training the models on more datasets could potentially bring some performance gains.

The multi-word model was able to leverage the datasets from both tasks, to identify two word phrases. This is typically a more challenging than the assessment of single words, due to the difficulty of properly representing MWEs. Including a part-of-speech feature may lead to some improvements with this model, by helping it deal with different MWE formations, for which the model shows different accuracy. Due to the architecture design and the dataset, this model is not able to assess phrases with more than two words. It would be useful to make predictions for phrases of an arbitrary length, which would require a different architecture.

Both models are able to assess the context of the target text when making predictions. Although, as the left and right contexts are given as an average, all words are weighted equally regardless of their relevance to the target text. Because of this equal weighting of words, the models are able to adjust their predictions based on the general complexity of the sentence but are unable to fully capture the relevant context. Consequently, long sentences with a large number of highly frequent words, such as conjunctions, will lose a lot of the important semantics when being assessed by the model. Adding a mechanism that could weight each word in the context based on certain features may offer some improvements in this area. However, other architectures, such as BERT (Devlin et al., 2019) are more suited to this approach, using an attention mechanism to produce embeddings that capture large amounts of contextual information. Fine tuning such a model for CWI, with the inclusion of features such as word frequency, could produce representations that contain more useful and relevant information.

5 Definition Modelling

Previous work on the task of definition modelling (DM) (Bevilacqua et al., 2020) has seen success with the use of natural language generation systems, with the clear benefit of not being restricted to a vocabulary of definitions. Such approaches however are susceptible to "hallucinations", causing the model to infer a incorrect definition from the context of the sentence, when lacking adequate knowledge of the target text. These hallucinations can appear to be rather convincing and have the potential to cause the user to mistake an incorrect definition for a real one. For this reason a retrieval based approach towards this task was taken based on the supposition that the model's restricted vocabulary may reduce the prevalence of convincing but false definitions, making it a more appropriate choice for the CoAST use case.

Rather than producing, or selecting simplified versions of text, through Lexical Simplification, a DM system has been chosen to produce the annotations in CoAST. Comprehensive definitions function as explanations which can provide readers with a more in-depth and robust understanding of a word or phrase. Additionally, some phrases, especially domain-specific academic terms, cannot be easily simplified on a word-to-word basis.

5.1 Methodology

5.1.1 Model Architecture

The model developed as part of this work, which will be referred to as DDR (Dense Definition Retriever), is a type of dense retrieval network and is used for the retrieval of definitions. Dense retrieval networks utilize deep neural networks to encode queries and documents in the same low-dimensional space (Luan et al., 2020; Xiong et al., 2020). This allows the desired document (in this instance a definition) to be retrieved using a nearest neighbours search between an encoded query and an index of pre-encoded documents.

DDR employs a dual-encoder architecture using two seperate BERT base (Devlin et al., 2019) encoders. The encoders E_{CT} and E_D are initialized using the weights of the question and passage encoders from the dense passage retrievers multiset checkpoint (Karpukhin et al., 2020). The model is trained with the objective of maximising the similarity between the encoded representation of a context—target (CT) pair and its respective definition (D). These representations are obtained by taking the output of the [CLS] token (the first token in the sequence) from the last layer of both encoders. The CLS token is generally referred to as the "pooled output" because it is used to denote the start of a sequence, and by the last layer it is able to capture the necessary context from the rest of the sequence due to self-attention. Similarity in this case is defined as the dot product between these two representations:

$$sim(CT, D) = E_{CT}(CT) \cdot E_D(D) \tag{2}$$

The BERT model is trained to recognise the [SEP] token as a denotation of a separate sentence as part of its pre-training objectives. In order to distinguish the text we want defined from its surrounding context this [SEP] token is utilized with CT pairs being passed to E_{CT} in the format: [CLS] Context [SEP] Target Text.

In line with previous literature on the training of dense retrieval networks (Karpukhin et al., 2020; Li et al., 2019; Xiong et al., 2020), negative sampling is used as a contrastive learning approach for the training of DDR. In terms of this task a negative sample is any CT pair or D that is not matched with its target counterpart. For the initial five epochs of training in-batch sampling is used to reduce computation by reusing every definition in the batch as a negative sample for every other item in the batch. This results in $BatchSize - 1$ negative samples for each item in the batch. Section 5.1.4 provides more details on the different negative sampling techniques used and the justifications for doing so.

After the outputs from both encoders have been obtained a batch-wise matrix multiplication is performed between $E_{CT}(CT)$ and $E_D(D)$ which provides dot products between each CT pair and its positive and negative samples. A softmax operation is then applied over the first axis of this output, which produces a probability distribution for the definitions given as samples for each CT pair. Negative Log Likelihood is used as the objective function, with the model aiming to maximise the likelihood of a positive example and minimize the likelihood of all negative samples through training. This process leads to vector space in which CT pairs are in close proximity to any appropriate definitions. The loss function that is used throughout the majority of training is depicted below (Karpukhin et al., 2020):

$$-\log \frac{e^{sim(CT_i, D_i^+)}}{e^{sim(CT_i, D_i^+)} + \sum_{j=1}^{n} e^{sim(CT_i, D_{i,j}^-)}} \tag{3}$$

5.1.2 Datasets

A comparative overview of each of the datasets is given in table 8. The majority of training is performed on the CHA dataset (Chang and Chen, 2019) which consists of CT pairs and definitions taken from `oxfordictionaries.com`. CHA features 79,030 unique definitions and a train/dev/test split. For the purposes of evaluation 1000 unique words are selected at random and removed from the dataset to form a testing dataset called CHA$_U$. The remaining dataset, which will be referred to as CHA$_S$, contains words and definitions in its test split that are present in the training split but are presented in a different context. CHA$_U$ consists of unseen words not present in the networks training data.

Additional training is carried out on the SEMCOR Word Sense Disambiguation training dataset (Raganato et al., 2017) using definitions retrieved from WordNet to construct another training corpus, which will be referred to as SEM$_{WSD}$. Words belonging to synsets containing less than 4 lemmas and words present in CHA$_U$ are removed from this corpus, leaving 152,799 examples with 13,302 unique definitions. Note this is a different set of data to the SEM dataset constructed by Bevilacqua et al. (2020) which utilized the full SEMCOR corpus (Miller et al., 1993)

HEI++ (Bevilacqua et al., 2020) is a dataset contructed for the evaluation of DM on standalone adjective-noun phrases. Despite only training on individual words HEI++ is utilized to test the model's ability to map phrases to definitions not present in the training corpora. Definitions in this dataset are handwritten to a high standard by an expert lexicographer.

	Instances			**Definitions**		
Dataset	**Train**	**Dev**	**Test**	**Train**	**Dev**	**Test**
CHA$_S$	538321	76016	146366	75663	31900	36188
CHA$_U$	—	—	24848	—	—	2521
SEM$_{WSD}$	152799	—	—	13302	—	—
HEI++	—	—	713	—	—	713

Table 8: Definition Modelling Datasets

5.1.3 Inference

In order to efficiently retrieve definitions at inference time the FAISS (Johnson et al., 2017) library is used to index all the definitions from the CHA dataset. FAISS is a library designed for performing similarity searches over large indexes of data and is able to scale efficiently to work with billions of vectors.

To process our data the HuggingFace datasets library[5] is used, which allows us to produce a dataset from a .csv file containing all our definitions with the columns embedding and text. This dataset is then iterated, processing each entry with the trained encoder E_D, and mapping the outputted vector to the column “embedding”. After all the definitions have been encoded an index optimized for approximate nearest neighbours search is applied to our dataset. Now the top-k nearest definitions can be retrieved for a given CT pair by passing the the output of $E_{CT}(CT)$ to our dataset and performing a search.

5.1.4 Sampling Techniques

This section will give an overview of the different sampling techniques used to obtain negative samples. The quality of negative samples used during training is key for maximising the convergence of retrieval models (Xiong et al., 2020). If a negative sample is too trivial to differentiate from the correct definition then the model will receive uninformative training signals, and therefore little learning will take place. Conversely, if a negative sample is too similar to the correct definition then there becomes an increased risk that the sample is a false positive (Li et al., 2019; Xiong et al., 2020). Essentially meaning that the use of the negative sample in the context of the CT pair would be equally as appropriate as the “correct” definition given by the dataset. This is problematic as the model may be penalized for attributing a high probability to a suitable definition, which would encourage over-fitting to the training set.

In-Batch Sampling: As specified in section 5.1.1 the in-batch sampling technique is employed for the first five epochs of training. Although this method is initially computationally efficient it becomes continuously less effective as training progresses. This is because as the model starts to adapt to the task the probability of an informative sample being present in the batch decreases. Lack of informative samples will lead to a decreasing gradient of the cost function which means the rate of improvement of the model will decrease. The other sampling methods discussed below are used in conjunction with this sampling method.

Hard Negative Sampling: Hard negative samples refer to samples that are difficult

[5] https://huggingface.co/docs/datasets/

for the model to differentiate from the gold definition. To continually select more challenging negative samples the FAISS (Johnson et al., 2017) library is used to select definitions that the model has placed in a similar vector space to each CT pair at its current stage of training. The proximity of definitions in vector space is supposed to represent their semantic similarity, however while the model is not fully trained there will be many inaccuracies in the arrangement of these vectors. By using these definitions as negative samples the model is able to learn a more appropriate placement of their vectors through the training signals it receives.

In a similar fashion to the method detailed in section 5.1.3 each definition in the dataset is passed to the encoder E_D and the resulting outputs are indexed through FAISS. The top-k definitions for each CT pair in the dataset are then retrieved to be used as negative samples. This process is continually repeated each epoch to retrieve samples that will be challenging for the model. Between 1 and 4 hard negative samples ($0 < k < 5$) were used for each CT pair, depending on the computing resources available.

During training hard negative samples for each CT pair are appended to the end of the batch of E_D; consequently, all positive and negatives serve as negatives for every other item in the batch resulting in a negative sample size per CT pair of:

$$Negatives = (BatchSize \times HardNegatives) - 1 \tag{4}$$

Despite this sampling method being effective we eventually run in the problem of false negatives being selected as hard negatives. These false negatives can either be completely synonymous or applicable but less specific than the target definition. This is a weakness of this method of negative sampling; however, this problem is more pronounced with the task of definition modelling where there are many applicable ways to define a word and there will often be some ambiguity between a word and a list of candidate definitions. Increasing the number of hard negatives used helps to temporarily delay this issue, presumably by reducing the impact of a false negative being present.

WordNet Sampling: While training on the SEM$_{WSD}$ corpus, the lexical database WordNet (Miller, 1995) is utilized to select additional definitions from the target words synset as hard negative samples. Synsets are groups of word senses that refer to similar concepts[6]. This sampling technique is used to help train the model to disambiguate

[6] https://wordnet.princeton.edu/

between different word senses and to partially avoid the issue of false positives as it is taken on assumption that WordNet senses are mostly distinct.

Hard Negative CT Sampling: Many definitions share a large lexical overlap, and as previously discussed there are many overly general definitions that could potentially be applied to multiple CT pairs. To try to combat this issue we experimented with a different training approach, marginalizing over positive and negative CT pairs for each definition in the batch. This process follows a similar methodology to the selection of hard negative samples, except CT pairs are indexed and retrieved rather than definitions. Hard negative CT sampling led to some improvements in terms of the validation loss however this improvement began to plateau after a few epochs. This sampling method is still prone to the retrieval of false positives. Additionally, it is a very computationally expensive process to perform every epoch as 538,321 CT pairs have to be processed by E_{CT} and indexed, compared to 75,663 definitions. The loss function used for this sampling method is detailed below in equation 5. This is the same loss function as stated previously in equation 3, except for the inversion of CT samples and definitions.

$$-\log \frac{e^{sim(D_i,CT_i^+)}}{e^{sim(D_i,CT_i^+)} + \sum_{j=1}^{n} e^{sim(D_i,CT_{i,j}^-)}} \tag{5}$$

5.1.5 Training Details

The model was trained for 30 epochs in total using the Adam optimizer (Kingma and Ba, 2014). For the initial 5 epochs the in-batch sampling method was used with a learning rate of $1e-5$. For the rest of training the learning rate was decreased to $1e-6$, using hard negative sampling for 15 epochs in total, and training for 5 epochs with each of the other methods. Because our system uses in-batch samples as a contrastive learning approach, larger batch sizes were more effective, this is demonstrated in previous work with the Dense Passage Retriever (Karpukhin et al., 2020). For the majority of epochs a batch size of 24 was used, however this was scaled up or down depending on the hardware available:

- **T4 GPU:** With this GPU a batch size of 12 was used with a hard negative sample size of 3. The in-batch sampling training was also performed on this device, using a batch size of 30. Training with the T4 took approximately 12 hours per epoch.

- **V100 GPU:** With this device a batch size of 24 was used, with 3 hard negative samples per item in batch. Training with the V100 took approximately 6 hours per epoch.

- **V3-8 TPU:** Some of the training was performed on a TPU. This allowed the batch size to be scaled to 72 with 4 hard negative samples per item in batch. Due to the TPU's ability at handling matrix multiplication, training was substantially quicker, taking 40 minutes per epoch.

As new negative samples were retrieved after each checkpoint an additional 40-60 minutes per epoch was added to the training process. The model was written and trained using the Pytorch Lightning[7] framework, which makes it easier to switch between TPU based devices and GPU's throughout training.

5.2 Results and Analysis

DDR's performance on this task is presented in comparison with the two most recent pieces of work on DM. Section 5.2.1 provides an in-depth explanation of the metrics that have been used.

The Chang model (Chang and Chen, 2019) is a retrieval based model that uses BERT's (Devlin et al., 2019) contextualised embeddings in conjunction with FastText (Bojanowski et al., 2016) word embeddings to map words in context to their respective definitions. The GEN model (Bevilacqua et al., 2020) takes a natural language generation based approach to DM by fine-tuning BART (Lewis et al., 2019) on multiple datasets to generate a definition from a CT pair. More details on these model and their approaches to this task are provided in section 2.3.

5.2.1 Evaluation Metrics

Average Precision at K (P@K) is a metric used to measure the percentage of correct definitions that are given in the top-k retrievals. Results for $K = \{1, 5, 10\}$ are used to compare retrieval precision against previous systems.

The BLEU (Papineni et al., 2002) metric measures the average precision of n-gram matches between the target reference (gold definition) and the selected candidate. The BLEU-4 implementation from the NLTK library (Loper and Bird, 2002) is used, which

[7] https://github.com/PyTorchLightning/pytorch-lightning

calculates scores for up to 4 n-grams using uniform weights. As many definitions share similar phraseology, such as “In relation to...”, BLEU may give high scores to candidates that are conceptually unrelated to the target definition.

ROUGE (R-L) (Lin, 2004) functions in a similar fashion to BLEU, but focuses on recall rather than precision. The longest sequence of words that appear in both the target definition and the candidate are used to calculate scores for this metric. ROUGE is therefore susceptible to many of the inaccuracies of BLEU.

METEOR (MT) (Banerjee and Lavie, 2005) improves on some of the problems with BLEU and R-L by treating words that share the same base form (the root of a word, without the addition of prefix or suffix) on WordNet as equal, and stemming the words that do are not match before evaluating them. The recall of word and n-gram alignment is measured by this metric.

BERTSCORE (BS) (Zhang et al., 2019a) is a more recent metric on which Bevilacqua et al. (2020) reported the results of their sytsem. Contextualised embeddings are utilized by BS to address many of the pitfalls of string, n-gram and heuristic based metrics. Pairwise cosine similarity and inverse document frequency are used to compute weighted recall scores between candidates and target definitions. BS is well-correlated with human judgements on machine translation and image captioning tasks, in comparison to previous metrics, and shows a robustness against adversarial data on the PAWS dataset (Zhang et al., 2019b,a). However, it is important to note that the accuracy of automated metrics varies depending on the task (Fabbri et al., 2021; Gehrmann et al., 2021).

5.2.2 Retrieval Precision

Task	Model	P@1	P@5	P@10
CHA$_S$	Chang (base)	63.3	74.0	77.1
	Chang (large)	62.4	73.2	76.3
	GEN-CHA	67.9	72.9	74.7
	GEN-UNI	55.5	63.1	65.8
	DDR	**74.1**	**93.5**	**95.2**
CHA$_U$	Chang (base)	2.3	7.4	11.4
	Chang (large)	2.5	8.2	12.4
	GEN-CHA	6.5	16.8	22.0
	GEN-UNI	7.4	18.0	23.8
	DDR	**18.9**	**45.0**	**56.4**
HEI++	DDR	**35.9**	**64.2**	**71.8**
	DPR	30.3	50.0	56.5

Table 9: Precision@K values, measuring the percentage of target definitions present in the top-k retrievals.

In terms of retrieval precision, DDR boasts impressive gains over previous work on DM. For each dataset we can see a continuous improvement between P@1 and P@10. This suggests that the definitions are sensibly arranged in vector space, and the selection of a target definition is non-stochastic. As the GEN models are not retrieval based systems Bevilacqua et al. (2020) encoded defintions from all datasets, using the Sentence-BERT model (Reimers and Gurevych, 2019), and employed a similarity based ranking strategy to compare retrieval precision on this task.

On the CHA$_S$ dataset we see a improvement of 18.1 percentage points at P@10 with a precision of 95.2%, for the most important result of P@1, DDR maintains a respectable precision of 74.1%. As stated in section 5.1.2, the CHA$_S$ test set is used to examine model performance on previously seen words, presented in a different context. With DDR training on 30,903 unique words, its ability to consistently retrieve target definitions on CHA$_S$ is encouraging. However, the objective of any machine learning model should be to apply knowledge that has been accrued throughout training to data with different characteristics. While words on this dataset are given in a different context, and often used in a different sense, a higher level of precision should be expected on this testing split.

When evaluating precision on unseen words in the CHA$_U$ split we continue to see significant gains, which speaks to DDR's ability to generalize to this data. This improvement can be seen across all K values, doubling the results seen from previous work, with the majority of target definitions being featured in the top-10 retrievals.

To test DDR's retrieval precision on the HEI++ dataset the similarity between $E_{CT}(CT)$ and its encoded target definition $E_D(D)$ is computed, and ranked against all definitions present in our index. Despite not being trained on phrases, or any of the definitions present in HEI++, results from this dataset are still reasonable, especially when compared to performance on CHA$_U$. Part of this precision is likely due to the lack of sense ambiguity displayed by phrases. Both the adjective and the noun in these lexical items offer the model further contextual clues about the meaning of the text we want defined. Nevertheless, from these results we can see that DDR is able to deal with phrases. The models ability to rank unseen target definitions above those seen in training, which it presumably has some bias towards, is also promising for the CoAST use case. Ideally it should be possible for teachers to add any new definitions they provide throughout the annotation process to DDR's index.

The Dense Passage Retriever (DPR) (Karpukhin et al., 2020) model has been used as a baseline for the HEI++ dataset. This model is trained for question-answering tasks, and was used to initialize DDR weights at the start of training. It is therefore a good benchmark to see how well DDR has adapted to the task of DM. HEI++ features standalone phrases, and the lack of context makes it possible to evaluate DPR on this dataset, as there is no way to specify the target of a sequence. The results from DPR were obtained by phrasing the input as "What is the definition of '*target phrase*'?", and otherwise following the same methodology as with DDR. DM could be viewed as a question answering task, so DDR's improvements over the baseline, which become more pronounced for higher values of K, are reassuring.

5.2.3 Automated Evaluation

Many of the top-ranked retrievals that are not listed as a target definition may still accurately define their CT pairs, to some extent. To measure the similarities between the top-1 retrievals and their target definitions we employ the remaining automated evaluation metrics discussed in section 5.2.1. Results on these metrics are given in table 10. The random baseline given by Bevilacqua et al. (2020) is recorded here to help illustrate some of the inaccuracies displayed by each metric. Definitions from each test set are randomly sampled for each CT pair to produce this baseline.

Task	Model	BLEU	R-L	MT	BS
CHA$_S$	Random	0.2	10.8	3.2	68.1
	Chang (base)	74.7	78.3	-	-
	GEN-CHA	76.2	78.9	54.8	93.0
	GEN-UNI	66.9	72.0	47.0	90.7
	DDR	**78.7**	**79.2**	**77.9**	**93.1**
CHA$_U$	Random	0.3	11.0	3.2	68.2
	Chang (base)	7.1	19.3	-	-
	GEN-CHA	8.1	28.7	12.7	76.7
	GEN-UNI	8.8	29.4	13.5	76.8
	DDR	**36.5**	**36.0**	**34.0**	**79.1**
HEI++	Random	1.6	12.7	0.4	73.4
	DPR	40.0	39.4	35.5	81.0
	GEN-UNI	6.3	26.3	15.1	78.9
	DDR	**44.6**	**44.6**	**41.7**	**83.2**

Table 10: Evaluation Results across various metrics.

Instances where DDR has selected the target definition will inevitably receive a perfect score, hence the models performance on CHA$_S$ is unsurprising given its precision of 74.1% on P@1. Recall scores from the GEN models on the BS metric suggest this system is able to perform at higher standard when given the ability to freely generate definitions. On the CHA$_U$ testing split we continue to see marginal to sizeable improvements, depending on the metric. DDR's results on this split give the impression that many of the selected definitions are still relevant, despite its retrieval precision of 18.9%.

The higher recall scores given by BS for the random baseline on HEI++ demonstrates that this metric is still prone to many inaccuracies. Definitions in HEI++ were all authored by one person so BS may be capturing some of the stylistic similarities between each definition. If we sample random definitions from all the different datasets, rather than just the test set that we are evaluating on, the BS value drops to 69.72.

The results on HEI++ show that DDR's top retrievals are rated higher than its baseline of the results from DPR. The difference in scores seems to mainly reflect the 5.7% difference in P@1 precision. GEN-UNI's results on this dataset display a similar pattern to those on CHA$_U$, which are both quite poor. Due to the limitations of the metrics, it is difficult to make a conclusive judgement regarding GEN-UNI's true performance on this dataset. In many respects GEN-UNI's results are impressive, despite the low scores, as it is not able to generate a perfect match. If DDR's perfect matches are removed from the evaluation it gets a much lower BS value of 73.8. Although this is not a fair comparison, with a much higher probability of completely incorrect definitions with all the perfect matches removed.

5.2.4 Discriminative Tasks

Results on the Word in Context (WiC) (Pilehvar and Camacho-Collados, 2019) and Word Sense Disambiguation (WSD) (Raganato et al., 2017) tasks are provided in table 11. Both of these tasks are designed to measure a models ability to disambiguate between different word sense, an important aspect of DM.

Model	WiC (%)	WSD (F1)
Chang (base)	68.6	—
GEN-UNI	**71.1**	**76.3/73.0***
DDR	66.8	69.9
Baseline	50.0	63.4

Table 11: Results for WSD and WiC tasks. *: 0-shot

WiC is binary classification task in which a model is presented with a target word in two different contexts and has to predict whether both instances refer to the same word sense. The baseline given for this task is a random one. For the prediction of binary labels a similar methodology to Chang and Chen (2019) is taken, to allow for an apt comparison.

The top 3 definitions for each WiC are retrieved by DDR and the label TRUE is outputted if a definition occurs in both sets. Results on this task are disappointing, demonstrating that despite a higher retrieval accuracy DDR struggles with sense disambiguation. The reported average human-level accuracy on this task is 80% (Pilehvar and Camacho-Collados, 2019).

The WSD task measures a models ability to predict the correct WordNet sense. The baseline given for this task is the Most Frequent Sense (MFS) heuristic. MFS predicts the highest occurring sense in the tasks training corpus for each word (Raganato et al., 2017). GEN-UNI's results are reported on for the ALL set, which DDR is evaluated on, and in a zero-shot setting. The zero-shot setting covers lemmas that were not featured in the SEMCOR corpus (Miller et al., 1993), which was part of GEN-UNI's training data.

On this task definitions for each of the target words lemmas are retrieved from WordNet and passed to E_D. A softmax operation is then applied over the dot product between the output of $E_D(D)$ and the encoded CT pair. This allows the model to discriminate between word senses, by selecting the word sense corresponding to the definition assigned with the highest probability. DDR's accuracy for the WSD task is relatively poor in comparison to the state-of-the-art results reported by GEN-UNI. Interestingly, if lemmas that are featured during training on the SEM$_{WSD}$ corpus are removed from this evaluation, we see a much higher F1 score of 79.2%. However, as words featuring less than 4 lemmas were not included in this corpus this further demonstrates that DDR does have trouble discriminating between words with many senses.

Results on both these tasks highlight an area of improvement for DDR. When tested on the WiC task, the GEN-CHA model which is trained solely on the CHA dataset displays a lower accuracy of 69.2%. Similar results are seen on the WSD task where the GEN-UNI model sees a 8 point increase in F1 score in the zero-shot setting, compared to their GEN-SEM model trained on one dataset. Indicating that training on wider selection of corpora would lead to an improved ability to deal with sense ambiguity. A sizeable portion of the retrieval inaccuracies produced by DDR are likely due to these sense misinterpretations. It is therefore plausible that advances here would incite significant gains on the precision of retrievals.

5.3 Future Work

Results from the evaluation demonstrate that the described methodology is an effective approach towards the task of DM. In particular, the gains seen with the retrieval of unseen words and phrases are promising. The performance seen on the discriminative tasks however, is somewhat disappointing. This section will discuss some potential changes to the training process that could potentially lead to some improvements with the model.

Self-Referential Definitions: Many definitions in the training data contain the target word which they are defining. For example, one of the definitions for Parmesan given in the CHA dataset is “of a dish cooked or served with parmesan”. These self-referential definitions are quite commonplace in dictionaries and are problematic for two reasons. Firstly, they often lack any information that would be useful to the reader, which is obviously undesirable. Secondly, they are likely damaging to the training process, as the model will learn to favour string matches over actual semantics. This is a form of shortcut learning, where a model learns decision rules that often work on the training data, but do not transfer well to real-world scenarios (Geirhos et al., 2020).

Shortcut learning limits a models ability to generalize, and may cause DDR to retrieve definitions containing the target word instead of more informative alternatives. The CHA training set features 36,432 definitions that contain the target word, which is 6.77% of the total data. Removing these examples from the training data would help prevent this type of shortcut learning from taking place, this would likely require retraining the model from scratch.

Datasets: Training on a wider range of datasets would lead to a performance increase. This can be seen in the results reported by Bevilacqua et al. (2020), where the GEN model shows sizeable gains on the discriminative tasks when trained on a wider selection of corpora. Seeing a wider range of examples helps models build more robust representations. Additionally, there are a limited number of definitions compared to the number of corresponding CT pairs. The CHA dataset contains 79,030 unique definitions, with SEM$_{WSD}$ featuring 13,302. When training with 3 hard negative samples, on the CHA dataset, E_D is exposed to 1,614,963 definitions per epoch. This may have led to E_D over-fitting to the definitions. Training on a wider corpora would also help alleviate this.

Sampling Techniques: Informative negative samples became increasingly tricky to consistently obtain towards the end of training. As the model relies on these samples to improve, better methods of obtaining them would lead more robust and accurate results.

Cross-encoders are a type of classifier, that are more effective than dual encoders due to their ability to perform attention over both sequences. They are much less efficient however, making them unsuitable to use when marginalizing over large numbers of candidates. For the task of Open-Domain Question Answering Qu et al. (2021) utilized a cross-encoder to filter out potential false negative when selecting samples to train their dual-encoder. This is done by only selecting negatives that are classified as incorrect by the cross-encoder, which due to its capability, are more likely to be true negatives. This is obviously quite a resource intensive method of training, however cross-encoders can also be used to improve precision at inference time by re-ranking candidates. A similar methodology could also be employed for the task of DM.

Data Augmentation could be used as a method of reliably producing negative samples. This can be done through the use of simple rules that change the meaning of a definition through the addition of words such as “not”, or by replacing key words with antonyms. Language models are generally bad at handling negation (Hosseini et al., 2021), which is important for many tasks including DM, so using negated samples as negatives could lead to some performance gains. Selecting data transformations that cause the smallest perturbations in vector space, whilst still changing the meaning of the definition, would be most effective.

Target Text	**Prediction**	**Gold**
Run	Organize and initiate a campaign or other course of action	Organize implement or carry out
Discourse	A written or spoken discourse expressing considered thoughts on a subject	A formal discussion of a topic in speech or writing
Plea	a serious urgent or heartfelt request	a request made in an urgent and emotional manner
Phase	The property of matter that is responsible for electrical phenomena existing in a positive or negative form	Each of the electrical windings or connections of a polyphase machine or circuit
Industrial	Relating to or characterized by industry	Of a disease or injury contracted or sustained in the course of employment especially in a factory

Table 12: *Incorrect* top-ranking predictions with low levels of uncertainty

Incorrect predictions for which the model displays high uncertainty are more likely to be be true negatives. One way of approximating model uncertainty is through the use of dropout at inference time, a technique known as Monte-Carlo (MC) Dropout (Gal and Ghahramani, 2016). This involves performing N forward passes with dropout enabled, and using the variation between results as a measure of uncertainty. For this task, this

Target Text	Prediction	Gold
Moorish	Looking or sounding bizarre or unfamiliar	Relating to or characteristic of the moors
Donnybrook	A race similar to the derby run elsewhere	A scene of uproar and disorder a heated argument
Corporatism	A complex system of beliefs	The control of a state or organization by large interest groups
Symptomatic	Lacking distinctive or interesting features or characteristics	Serving as a symptom or sign especially of something undesirable
Aground	Of wind or tide opposed to ones desired course	With reference to a ship on or on to the bottom in shallow water

Table 13: Incorrect top-ranking predictions with high levels of uncertainty

could either be the variation in SoftMax probability or the variation in rank for a given definition.

It is also possible to approximate uncertainty by measuring the disagreement between an ensemble of multiple models. MC Dropout actually works in a similar fashion. During training when dropout is used, different sets of parameters are exposed to the data each epoch. When MC Dropout is used at inference time, it can show the disagreement between these different parameter sets. This process can also help detect an out-of-distribution sample, where a CT pair vector can be positioned equally, very distant, from a large number of definitions, hence small perturbations to the vector will cause a complete shift in the retrieved definitions.

By only selecting negative samples that display a high degree of variation, many false negatives could be avoided. Standard Error (SE) and Coefficient of Variation (CV) would likely be good measures of quantifying such variation. Examples of top-ranked *incorrect* model predictions, rated with low and high levels of uncertainty are given in tables 12 and 13. These were obtained from the CHA$_U$ dataset, by performing 10 forward passes over each instance with a dropout of 0.1 applied to each layer of E_{CT}. SE and CV values were obtained by recording the variation and mean values of the SoftMax probabilities of each prediction, and predictions were ranked by their mean value. Samples with a SE of less than 1.0 and CV of less than 30.0 are classed as low uncertainty. Recent work has suggested that it is sufficient to only apply dropout to the final layer of the network, which would reduce the computational requirements of such an approach (Cohen et al., 2021).

A large proportion of these low uncertainty predictions are false negatives, as they are applicable to the target word and similar to the gold definition. The examples given in the tables have been selected randomly from the results. Without manually labelling false positives in the dataset it is not possible to produce a more quantitative analysis, which would allow for less arbitary threshold values for SE and CV to be selected. Additionally, if accurate threshold values were obtained this technique could be employed at inference time, to re-rank predictions based on uncertainty, potentially improving the results (Cohen et al., 2021; Penha and Hauff, 2021).

6 Coast System

The main contributions of this work, in the tasks of CWI and DM, have been previously discussed and evaluated independently of the CoAST system. In this section these technologies will be integrated into CoAST. Additionally, we cover changes that have been made to CoAST, to resolve the prior issues discussed in section 3.3. An online study is conducted to evaluate the CoAST platform with the addition of this work's contributions. The code for CoAST is available on GitHub[8]

6.1 Methodology

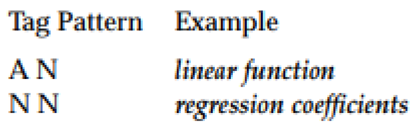
6.1.1 Highlighting System and Parsing

A new highlighting and parsing system has been added to address some of the issues discussed in section 3.3. The requirements of this system are to allow text of an arbitrary length to be highlighted, and to allow for complex words to removed when the difficulty level is decreased. As DIVs used for highlighting text change the structure of the document, a copy of the original content needs to be maintained.

A Class named highlightManager takes the document as input and builds an array containing JSON objects for each character. This JSON object contains any additional formatting that came before the character, whether the character is highlighted, fields to store DIVs that start or end highlights and the character itself. The document is split into characters; to avoid edge cases where there are double spaces or a new line between words that stop them being highlighted these characters are stored in the formatting attribute of the next character and their JSON object is removed. A copy of the documents text is obtained by mapping all the character attributes from the array of JSON objects to a new array, and joining them all into one string which is stored as the document property in the Class. All this allows the document to be rendered with all its formatting and highlights, while storing an additional copy of the document where each character's index corresponds to an element in the array.

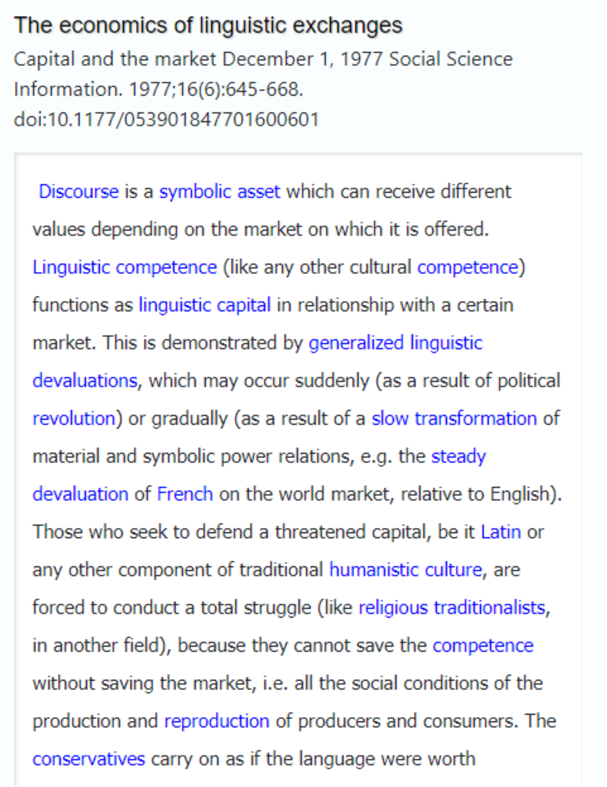
When text that needs to be highlighted is sent to highlightManager, the indexes for each position where it occurs in the document property are found. DIVs denoting highlighted text are then added to the character objects at the start and end of each words, and the document is rendered by combing all attributes in our object array into a string

[8] https://github.com/MMU-TDMLab/CTAS



Tag Pattern	Example
A N	*linear function*
N N	*regression coefficients*

Figure 9: Collocation filter used for identifying phrases



The economics of linguistic exchanges

Capital and the market December 1, 1977 Social Science Information. 1977;16(6):645-668. doi:10.1177/053901847701600601

Discourse is a symbolic asset which can receive different values depending on the market on which it is offered. Linguistic competence (like any other cultural competence) functions as linguistic capital in relationship with a certain market. This is demonstrated by generalized linguistic devaluations, which may occur suddenly (as a result of political revolution) or gradually (as a result of a slow transformation of material and symbolic power relations, e.g. the steady devaluation of French on the world market, relative to English). Those who seek to defend a threatened capital, be it Latin or any other component of traditional humanistic culture, are forced to conduct a total struggle (like religious traditionalists, in another field), because they cannot save the competence without saving the market, i.e. all the social conditions of the production and reproduction of producers and consumers. The conservatives carry on as if the language were worth

Figure 10: Example of CWI within CoAST

which is applied to the page. If the difficulty level is changed then the our character object array can be reset to remove all the highlights, and new words can be highlighted.

6.1.2 Integration of Complex Word Identification

The CWI system takes into account each word's context, so the text from our document needs to be split into sentences. Punctuation (excluding hyphens) is then removed from each sentence. To identify phrases, the part-of-speech based collocation filter displayed in figure 9 is used (Manning and Schutze, 1999). Although more comprehensive methods do exist, this method was chosen as a quick and simple way of identifying phrases. A library named en-pos[9] is used to label parts-of-speech.

Each sentence is split into strings of words, and phrases are identified and joined into one string. These strings are then used to create instances of a Class called WordManager, which is responsible for storing each string and its features. Section 4.1.2 describes

[9] https://github.com/FinNLP/en-pos

the features required by our model. Features for each word are retrieved and used to create an array with a length of 107. For strings that have been identified as phrases each WordManager instance creates two child instances which lookup and store features for each word individually.

The JavaScript implementation of TensorFlow is used to load the CWI model on the Node.js backend. To obtain complexity predictions each sentence and its instances of WordManager are iterated. For each piece of text the features corresponding to words to the left and right hand side of the target text are combined with the targets features to form a tensor which is passed to our model. Once the complexity values for each word/phrase are retrieved the results are passed back to the client where they are highlighted. Figure 10 gives an example of a document that has been processed by this system and highlighted for the user, on the beginner difficulty.

6.1.3 Integration of Definition Modelling

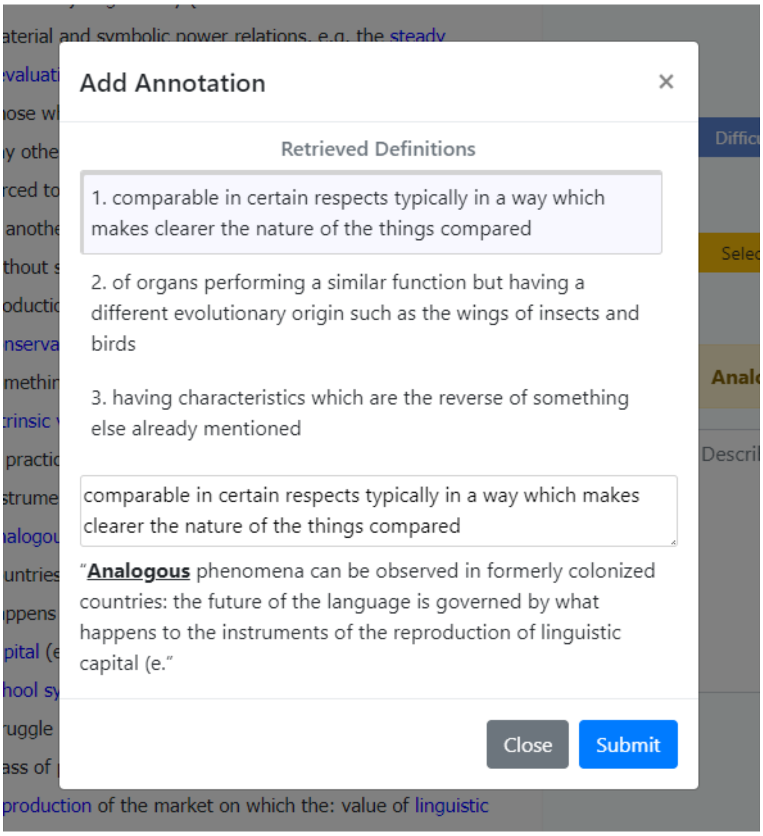
Interface

Definition modelling has been integrated into the annotation system in a way that is designed to enhance the teacher's user experience. Many definitions retrieved by the model will be appropriate but the teacher may still want to edit them to better suit the text, or to simplify some of the words within the definition. For each query the top three definitions are retrieved for the teacher to view, this increases the probability that a valid definition is present, without overloading the teacher with too many choices.

An example of the DM system within CoAST is provided in figure 11. When the teacher selects some text on the page and clicks the "Selected Text" button a popup containing the definitions appears. The sentence containing the target word is taken from the document and provided in the popup, to give the teacher some context while selecting a definition. When a definition is selected it is inserted into the text box, allowing the teacher to make any changes before submitting the annotation.

Implementation

The DM system is ran within a Flask server which allows us to run Python Code. The Node.js backend can communicate with this server through GET and POST requests. When a user selects some text on the page and clicks the "Selected Text" button, the text and its index within the DIV containing the documents text is retrieved. This positional index is used to obtain the surrounding context from the target text, which will be needed



1. comparable in certain respects typically in a way which makes clearer the nature of the things compared

2. of organs performing a similar function but having a different evolutionary origin such as the wings of insects and birds

3. having characteristics which are the reverse of something else already mentioned

comparable in certain respects typically in a way which makes clearer the nature of the things compared

"**Analogous** phenomena can be observed in formerly colonized countries: the future of the language is governed by what happens to the instruments of the reproduction of linguistic capital (e."

Figure 11: Example of Definition Modelling within CoAST

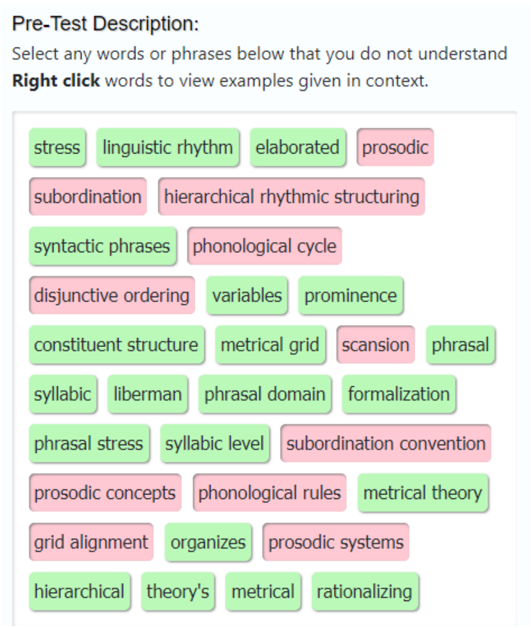
by the model. A JSON object is created to store the text and its context, which is sent to the backend server, and then to the Flask server. The text is first tokenized, then inputted to the model and the top 3 definitions are retrieved and sent back to the client. As shown in figure 11 these definitions are then displayed to the user in a pop-up. The context that was obtained earlier is provided at the bottom of this popup for the user's convenience.

6.1.4 Experimental Design

In order to assess the CoAST system a study will be conducted. This study will look at whether the models that have been integrated into CoAST aid the teacher in identifying complex words and writing annotations, and whether these annotations improve the participants understanding and vocabulary. To gauge each participants prior knowledge a pre-test will be conducted. The test will involve reading a document that will be presented with, or without annotations, followed by a post-test in which participants will match words to definitions given in a list.

There will be 4 sets of tests involving different methods of annotating the document. One where the teacher provides annotations without any assistance (referred to as teacher), and another where the teacher is able to utilize the DM and CWI system while writing annotations (assist). There will also be a test where the annotations are provided solely by the automated system without the teacher (auto), and a control test, where there is no annotations (control). Rather than assigning participants into groups, each participant will conduct all 4 tests, as there may be a limited number of participants available.

The test environment for the study has been integrated into the CoAST system and allows for the creation of tests by teachers or admins, and implements the test for students to take. For the purposes of the study, tests for 4 different documents have been added, which have been annotated with the methods discussed previously. After each test is completed, an entry is saved to the database, recording the text selected as difficult in the pre-test and the answers selected during the post-test by the participant.



Pre-Test Description:

Select any words or phrases below that you do not understand

Right click words to view examples given in context.

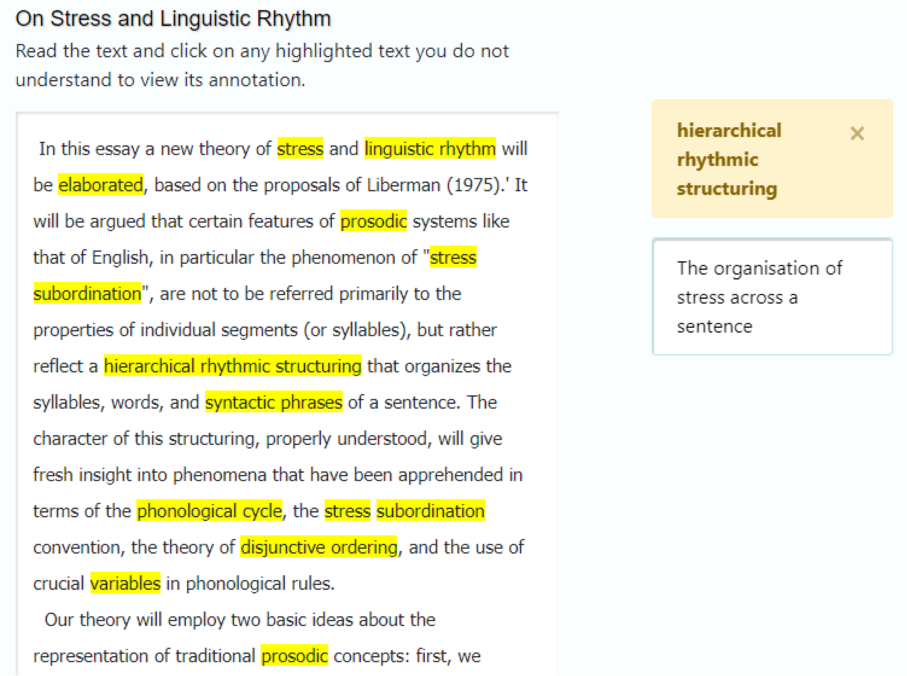
stress | linguistic rhythm | elaborated | prosodic
subordination | hierarchical rhythmic structuring
syntactic phrases | phonological cycle
disjunctive ordering | variables | prominence
constituent structure | metrical grid | scansion | phrasal
syllabic | liberman | phrasal domain | formalization
phrasal stress | syllabic level | subordination convention
prosodic concepts | phonological rules | metrical theory
grid alignment | organizes | prosodic systems
hierarchical | theory's | metrical | rationalizing

Figure 12: Pre-Test example

For each document the CWI system is able to select up to 1.5x the number of text items selected by the teacher. Words and phrases are selected by complexity in descending order, and text with a predicted complexity of less than 0.3 is excluded. The 1.5x limit is given, as setting an arbitrary complexity threshold could cause a large amount of terms to selected for certain documents, making the test process arduous for the participants. Selecting 50% more complex words helps account for: proper nouns that have been incorrectly selected; words which are the same but with morphological alterations; and, “pseudo” phrases that have been incorrectly identified by the collocation filter and hence will not be included in the post-test. For the document annotated in auto mode, the DM model relies on the words selected by the CWI model, hence the additional words also help prevent inaccuracies in the words selected affecting the DM system.

Pre-Test

For the pre-test each participant is asked to select any words or phrases from a list, that they do not understand. This list is produced by taking all the words that the CWI system, and the teacher have highlighted as complex when producing the test. An example of the pre-test for one of the document's is displayed in Figure 12. Right clicking any of the words or phrases produces a popup, providing the participant with sentences from the document containing the text.



On Stress and Linguistic Rhythm

Read the text and click on any highlighted text you do not understand to view its annotation.

In this essay a new theory of stress and linguistic rhythm will be elaborated, based on the proposals of Liberman (1975).' It will be argued that certain features of prosodic systems like that of English, in particular the phenomenon of "stress subordination", are not to be referred primarily to the properties of individual segments (or syllables), but rather reflect a hierarchical rhythmic structuring that organizes the syllables, words, and syntactic phrases of a sentence. The character of this structuring, properly understood, will give fresh insight into phenomena that have been apprehended in terms of the phonological cycle, the stress subordination convention, the theory of disjunctive ordering, and the use of crucial variables in phonological rules.

Our theory will employ two basic ideas about the representation of traditional prosodic concepts: first, we

hierarchical rhythmic structuring ×

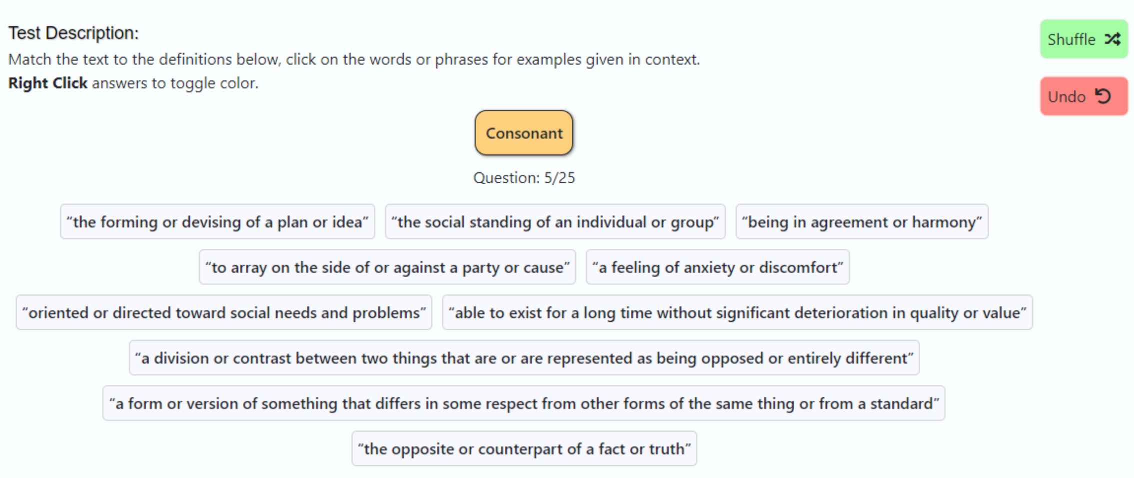
The organisation of stress across a sentence

Figure 13: Test example

Test

Figure 13 shows an example of one of the tests. While reading the document, the participant is able to click any of the highlighted text and view the annotations given. Each document contains text taken from linguistic research papers. Linguistic papers were selected as we have access to a teacher with linguistic expertise to perform the annotation process. Additionally, it is unlikely that the participants will be familiar with all the terminology used in these papers, as many of them will be from a computer science background. It is important that the selected documents are challenging for the participants because It will make it easier to measure the affect of each annotation process.

Post-Test



Test Description:

Match the text to the definitions below, click on the words or phrases for examples given in context.

Right Click answers to toggle color.

Question: 5/25

“the forming or devising of a plan or idea”

“the social standing of an individual or group”

“being in agreement or harmony”

“to array on the side of or against a party or cause”

“a feeling of anxiety or discomfort”

“oriented or directed toward social needs and problems”

“able to exist for a long time without significant deterioration in quality or value”

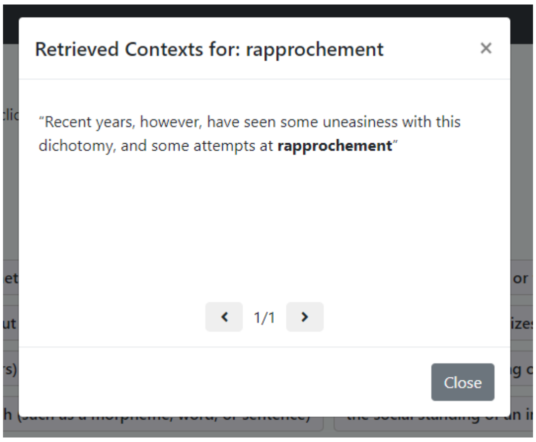
“a division or contrast between two things that are or are represented as being opposed or entirely different”

“a form or version of something that differs in some respect from other forms of the same thing or from a standard”

“the opposite or counterpart of a fact or truth”

Figure 14: Post-Test example

As shown in figure 14, in the post-test the participant is required to select a definition from a list for each question. The same words and phrases used in the pre-test are also used here, allowing us to gauge the impact that reading the annotations has had on the participants vocabulary. From the control document, with no annotations, we should also be able to see the extent at which a participant selecting a word they do not know in the pre-test, correlates to selecting the incorrect definition in the post-test.



Retrieved Contexts for: rapprochement

“Recent years, however, have seen some uneasiness with this dichotomy, and some attempts at **rapprochement**”

Figure 15: Contexts taken from the document for the selected word for the participant to view

For each question/word the number of definitions to choose from is limited to ten, as choosing from a larger number would become too challenging and time consuming. Sentences containing each target word are displayed to the user upon clicking its button, to provide the participant with some context when making their decision. These sentences are displayed in a popup which is shown in figure 15.

6.2 Results and Analysis

In total 10 participants took part in the study completing 39 tests across the 4 documents (teacher, assist, control and auto). Participants were able to complete the tests for each document independently; one of the participants failed to complete the tests for the teacher document within the given timeframe. A box plot showing the distribution of the average percentage of words selected by the participants as difficult in the pre-test is given in figure 16.

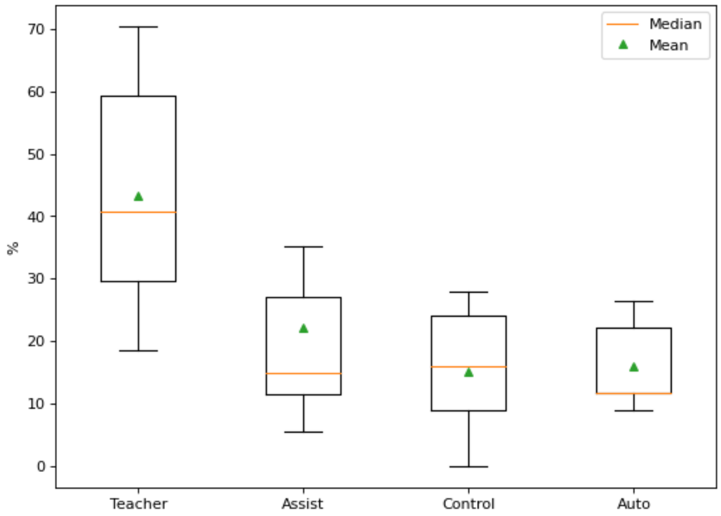


Figure 16: Box plot showing the percentage of words selected as difficult by the participants

The teacher document was perceived to be significantly more difficult than any of the others, with participants selecting 43.21% of the words on average. It is not possible to reliably account for this difference in text difficulty, as the number of participants is relatively small. Hence, this document has been excluded from the rest of the analysis. One of the participants did not complete the test for this document,

Document	Words Selected in Pre-test	Correct Answers
Assist	22.16	82.98
Auto	15.88	81.76
Control	15.20	78.40

Table 14: Percentage of words selected in the pre-test and the average percentage of correct answers, on each document

From table 14 we can see that the participants performed worse on the control document. The percentage of words selected in the pre-test is given for comparison in this table. Participants perceived the assist document as slightly more difficult than the others, although figure 16 shows that there is a significant overlap between the pre-test results

Document	Correct Given Selected	Correct Given Not Selected
Assist	65.58	85.84
Auto	67.08	84.48
Control	58.71	80.24

Table 15: The percentage of words that participants selected, or did not select, in the pre-test, that they got correct in the post-test

across each of the documents. Despite the average percentage of pre-test selections being higher on the assist document, the post-test results are not effected, suggesting that the participants performance may have mediated by the annotations.

Table 15 shows that participants are more likely to incorrectly match a word to a definition in the post-test, if they selected that word as difficult in the pre-test. Therefore, the percentage of pre-test selections can be seen as an indicator of text difficulty. Participants showed a lower accuracy on the control document in the post-test, this is further pronounced for words that they perceived to be complex.

Document	Overlap (%)	Number of Teacher Selections
Assist	88.46	25
Control	70.59	15
Auto	47.06	17

Table 16: The percentage of words highlighted as complex by the teacher, that were also highlighted by the model. The total number of words or phrases selected by the teacher as complex is given on the right column

Document	Correct by Teacher	Correct by Auto
Assist	80.80	83.14
Control	75.33	79.52
Auto	75.88	86.40

Table 17: The percentage of words that were highlighted as complex by the teacher, or by auto, that the participants got correct

From table 17 we can see that participants on the assist document show around a 5 percentage point increase in accuracy over the other documents, for words that were highlighted as complex, and annotated by the teacher. Assuming that the words highlighted by the teacher were of a similar difficulty on each of the documents, this suggests that participants performed better when presented with annotations written by the teacher using the assistive models.

Participants attained more correct answers for words that were highlighted as complex by auto on all of the documents. The CWI model is assumably less accurate than the teacher at identifying complex words, which is likely the reason for this effect. Participants got more answers correct on the auto document, when they were provided with the models annotations. From table 16 we can see that there is a much lower overlap between the teacher's, and model's selections, on the auto document. While the annotations given by the model may have had some effect on the participants performance, this improvement is likely because the model selected easier words on this document.

The teacher identified a greater number of complex words when provided with the models predictions in table 16. Additionally, there is a larger overlap between the teachers, and the models selections on the assist document.

6.2.1 Sub Analysis

It would be desirable if DDR is able to interpret new definitions provided by the teacher, and apply them in different contexts. This was tested to an extent in section 5.2.2 with the HEI++ dataset, however this only included examples for standalone phrases, real-word examples with context are likely to be far more ambiguous.

To measure the capability of DDR at generalizing to definitions outside of its training distribution, definitions provided for the experiment have been ranked against the entire index of definitions used by DDR. As the number of annotations used in this analysis is relatively small, the results cannot be taken conclusively, although it is useful to see the general trends in the data. Definitions provided by the teacher are encoded by E_D, and a dot product is performed against $E_{CT}(CT)$. This is then ranked against the dot products of all the definitions returned by DDR. The results are shown in table 18.

Mode	P@1	P@3	P@10	P@20	Total Annotations
Teacher	12.5	18.75	31.25	43.75	15
Assist	34.62	76.92	80.77	84.62	25

Table 18: P@K values for definitions provided by the teacher, when ranked against the entire index by DDR

For many words, there may be a selection of perfectly suitable definitions definitions in DDR's index. Hence, low precision's for P@1 or P@3 are not necessarily bad, however the definition provided by the teacher should at least be within the top 20 retrievals. Definitions that are not within this range either represent a misinterpretation of the context and target word, or a misinterpretation of the teacher's definition. Values for P@3 are given as this is the number of results that are retrieved for the teacher when using CoAST.

Results are reported for the document annotated by the teacher with assistance from both models (Assist), and the document annotated solely by the teacher (Teacher). In the Assist mode the teacher is able to select definitions that are provided by DDR, if they are appropriate. Therefore, as many of the definitions given may be taken directly from DDR the retrieval precision is higher on this annotation mode. This does show, for this particular document, DDR is either able to retrieve an appropriate definition, or correctly interpret a custom definition, 76.92% of the time for P@3. It can therefore be assumed that the majority of the definitions shown to the teacher were of relevance.

On the document annotated by the teacher, when the retrieved definitions are factored out, we can see much lower precision. This means that DDR is not able to reliably retrieve a custom definition. Only 18.75% of the teachers definitions are featured within the top-3 retrievals. However, from the Assist mode we can see that the majority of the definitions for P@3 are probably relevant anyway. For P@20 there is a precision of 43.75%, suggesting that just over half of the time DDR will misinterpret the given definition, or the target text. From viewing the retrievals it seems that DDR will rank a custom definition higher, if it shares words, or synonyms of words to that of its top ranked retrievals. This is to be expected, but the results do suggest that E_D is somewhat overfitted to the definitions it has been exposed to during training.

7 Discussion

This thesis has explored the use of machine learning based tools within the context of a VLE, aiming to improve reading comprehension and language acquisition through their application. The findings and contributions from the work on CWI and DM are summarized as follows:

1. **Complex Word Identification**

   The task of CWI is investigated in this work. It may not be immediately apparent to a teacher using a system such as CoAST, which words, or phrases, their students might struggle with. This portion of the project implements two convolutional based neural networks, trained to identify complex words and two-word phrases, in a context-dependant manner.

   The results from the evaluation on the CompLex dataset (Shardlow et al., 2020) indicate our approach is reasonably effective. From the study conducted in section 6.2 we can see that the models predictions for complex words share a sizeable overlap with the teachers selections. On the assist document where the teacher is able to view the highlighted predictions, more words are annotated by the teacher, and a larger overlap of 88.46% is seen with the CWI models predictions. These results suggest that the CWI models improve the teachers ability to identify complex terminology and, are a helpful addition to the CoAST system.

2. **Definition Modelling**

   A dense retrieval network is trained to retrieve definitions based on a word or phrase given in context, this model is referred to as DDR. Definition modelling is applied to the CoAST system to assist the teacher in writing annotations.

   Results from the automated evaluation demonstrate the effectiveness of our approach. In particular, the model shows sizeable gains over previous work when dealing with words not seen during training. DDR displays worse performance than other work at discriminating between word senses, when evaluating on the tasks of WSD and WiC. In the study conducted involving CoAST the document annotated by the teacher with the assistance of the DM model saw slightly better results. From the sub analysis (section 6.2.1) we can see that DDR is able to retrieve a relevant definition for the teacher the majority of the time. DDR shows a relatively low precision when retrieving definitions given by the teacher, suggesting that adding new definitions to the models index may not be effective due to overfitting.

This project as a whole looked at using machine learning tools to assist the teacher with the identification of complex terminology, and with the writing of helpful annotations within CoAST. The aim of the CoAST system is to improve the student's reading comprehension, and encourage language acquisition, it is therefore important that the contributions of this project further this goal.

Both the CWI and the DM models were evaluated independently of CoAST, with the results showing that these technologies work effectively at the tasks they were trained for. Research on NLP for education is often evaluated through various benchmarks, and comparisons with human gold-standard labels, without assessing its functioning in a educational environment (Litman, 2016).

The online study in section 6.2 was conducted to extrinsically evaluate the NLP technologies produced in this work within the context of the CoAST system. Results from this study suggest that the assistive technologies integrated into CoAST (CWI and DM) allow the teacher to identify, and annotate, more complex terminology in a document. When using the assistive features the teacher provided more annotations, many of which were suggested by DDR, and the words selected as complex shared a large overlap with the CWI models suggestions. Consequently, it is likely that the application of these technologies does indeed improve the ease of use of CoAST and the annotation process. However, further evaluation with more teachers using the system would be useful to better understand the degree to which the assistive technologies benefit the teacher.

Results from the analysis with CoAST suggested that the participants perceived the assist document to be slightly more difficult than the others in the pre-test. Despite this, participants on the assist document attained more correct answers on average in the post-test, for words which were provided with annotations. This suggests that the CoAST system would be beneficial for students, and their comprehension of difficult texts. However, this improvement was not statistically significant given the number of participants.

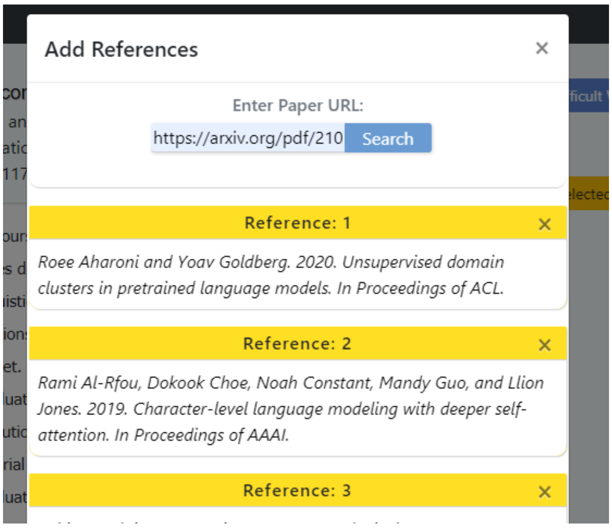
The words that were featured in the pre- and post- test are those that were selected as complex by the teacher or the model. Consequently, the participants accuracy on the post-test, and the percentage of selections in the pre-test, will have been mediated by the accuracy with which complex words were identified in each document. The teacher's selections of complex words will likely be more consistent than the models across the different documents. If the words selected by the CWI model are more understandable on one document, then participants will do better on that document. This is a flaw in the test design. Only using the teachers selections for the post-test, and annotating a random selection of them, would have made it easier to isolate the effect of each annotation mode, and the difficulty of each document.

The difference in results between each document may have been more pronounced with a different experimental format. Multi-choice questions were used for the post-test, which would potentially allow participants to deduct the correct answer from knowledge of the other definitions they have to choose from. This may lead to a cumulative effect where participants will get more of the words they consider difficult correct, the easier they find the document as a whole, and the less definitions they have to actually choose from. Some of the errors made by participants are due to selecting an incorrect definition, that is similar to the target one. For example, in the Assist document the definitions for bilingualism, bi, multilingual experience, multilingual identities, multilingualism and multilinguals were often confused with each other. Documents where participants have to choose from lots of similar terms, will likely see more incorrect answers. A free form answering style post test would be a better approach, which would have avoided these issues, however this would be more resource intensive, requiring the marking of 1203 answers.

It is difficult to assess the effect that the annotations on the auto document, produced solely by DDR, had on the participants results. The participants found the words selected as complex by the CWI model easier than the teacher's selections, on all of the documents. Additionally, it seems that the model identified complex words with less accuracy on the auto document (the models predictions shared a lower overlap with the teachers). It is therefore not possible to tell whether the participants performed well because of DDR's annotations, or because the CWI model selected easier words. Presenting definitions selected by DDR for the teacher's complex word selections, rather than the CWI model's, would have been a better approach, allowing the auto and assist documents to be directly compared.

7.1 Future Work

When annotations are added to a term within CoAST they are displayed for every instance of the term within the document. This poses is a problem if a document contains two senses of a word and the teacher wants to provide different annotations for each of them. Additionally, the CWI system provides the client with the same complexity predictions for every instance of a given term in a document. Individual predictions for each word instance are produced by the model, however, these are averaged, and it is this average that is used by CoAST. Both of these issues would require further changes to the highlighting system implemented in CoAST, to allow different instances of words to be treated differently. Changes to the way annotations are stored within the database would also have to be made, to reflect the index of their associated words in the document. Ensuring that the user interface of CoAST is not over complicated by allowing for this functionality would be important, and potentially challenging.



Enter Paper URL: https://arxiv.org/pdf/210

Reference: 1

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In Proceedings of ACL.

Reference: 2

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. 2019. Character-level language modeling with deeper self-attention. In Proceedings of AAAI.

Reference: 3

Figure 17: Interface for adding references to a document in CoAST

Academic documents generally contain many in-line citations, which are often highlighted as complex terms in CoAST. This can be problematic if a document contains many citations which are highlighted, detracting from any legitimately highlighted complex words. Rather than train a model to identify, and exclude these citations, it would be useful to provide users with a link to the work that the citation is referencing. As part of this work the interface for retrieving references was implemented within CoAST, which is displayed in figure 17. This uses an API[10] to retrieve a list of references, and their URL's, when given the URL of a PDF document. The additional functionality off identifying these citations within the document, and providing links to them has not been implemented.

The index of used by DDR is limited to definition's belonging to individual words, although some of these may also be applicable to various phrases. Additionally, Some of the definitions in the index may also be of poor quality, or contain complex terminology.

[10] https://ref.scholarcy.com/api/

When a teacher adds a new definition for a word or phrase in CoAST it would be useful if this definition could be retrieved by DDR for a similar word in the future. From experiments conducted in this work it seems that the model has a limited ability to generalize to new definitions. Experimenting with different sampling techniques and expanding the training corpora of the model would likely help reduce this overfitting.

8 Conclusion

The CoAST platform is designed to enhance the students reading experience, improve reading comprehension, and encourage language acquisition, by enriching theoretically, and lexically complex texts with the teachers pedagogical knowledge. This thesis has investigated machine learning models, that can be applied within CoAST to assist the teacher in identifying, and annotating, complex and potentially unfamiliar terminology. We reached the following conclusions:

- Words and phrases identified by our CWI model shared a sizeable overlap with terms the teacher also considered to be complex. The model was assessed as part of the Lexical Complexity Predication 2021 shared task, with results in comparable range to the other submissions. Results from this work support its use in CoAST and suggest that the teacher is able identify more terms that may be difficult for students, when exposed to the models suggestions.

- Our approach to the task of definition modelling is effective, able to retrieve definitions for words with a precision of 74.1%, and handle arbitrary length phrases. When used within CoAST this model can function to assist the teacher with writing annotations, by suggesting suitable definitions. The sub-analysis in section 6.2.1 demonstrated that the majority of retrieved definitions were useful for the teacher.

- Results from our study with CoAST suggest that these technologies would benefit teachers, and students, using such a system. The generalizability and reliability of our results is limited by the sample size. Additionally, a different experimental design may be needed to eliminate potentially confounding variables. This thesis contributes to work on the development of VLEs and demonstrates that machine learning technologies have the potential to, enhance the usability of these systems and, collaborate with teachers to improve the learning experience that VLEs provide.

References

Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. 2014. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22(10):1533–1545.

Terry Anderson and D. Randy Garrison. 1998. Learning in a Networked World: New Roles and Responsibilties. In *Distance Learners in Higher Education: Institutional responses for quality outcomes*, Atwood Publishing.

Segun Taofeek Aroyehun, Jason Angel, Daniel Alejandro Pérez Alvarez, and Alexander Gelbukh. 2018. Complex word identification: Convolutional neural network vs. feature engineering. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, New Orleans, Louisiana, pages 322–327.

Snjeana Babi. 2012. Factors that Influence Academic Teacher's Acceptance of E-Learning Technology in Blended Learning Environment. In Adilson Guelfi, editor, *E-Learning-Organizational Infrastructure and Tools for Specific Areas*, InTech.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Association for Computational Linguistics, Ann Arbor, Michigan, pages 65–72.

Tuqa Bani Yaseen, Qusai Ismail, Sarah Al-Omari, Eslam Al-Sobh, and Malak Abdullah. 2021. JUST-BLUE at SemEval-2021 task 1: Predicting lexical complexity using BERT and RoBERTa pre-trained language models. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*. Association for Computational Linguistics, Online, pages 661–666.

F. Behjat, M. Yamini, and M. Bagheri. 2012. Blended learning: A ubiquitous learning environment for reading comprehension. *International Journal of English Linguistics* 2:97.

Richard Bellman. 1957. *Dynamic Programming*. Dover Publications.

Christian Bentz and Ramon Ferrer-i Cancho. 2016. Zipf's law of abbreviation as a language universal. In *InProceedings (Aufsatz / Paper einer Konferenz etc.)*.

Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or "how we went beyond word sense inventories and learned to gloss". In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, pages 7207–7221.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. 2016. Enriching word vectors with subword information. *CoRR* abs/1607.04606.

Marc Brysbaert. 2012. crr " age-of-acquisition (aoa) norms for over 50 thousand english words. Available at `http://crr.ugent.be/archives/806`.

George C. Bunch, Aída Walqui, and P. David Pearson. 2014. Complex text and new common standards in the united states: Pedagogical implications for english learners. *TESOL Quarterly* 48:533–559.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. *Proc. of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology* .

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *CoRR* abs/1803.11175.

Ting-Yun Chang and Yun-Nung Chen. 2019. What does this word mean? explaining contextualized embeddings with natural language definition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, pages 6064–6070.

Daniel Cohen, Bhaskar Mitra, Oleg Lesota, Navid Rekabsaz, and Carsten Eickhoff. 2021. Not all relevance scores are equal: Efficient uncertainty and calibration modeling for deep retrieval models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, New York, NY, USA, SIGIR '21, page 654–664.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann LeCun. 2016. Very deep convolutional networks for natural language processing. *CoRR* abs/1606.01781.

Fred D. Davis. 1985. *A technology acceptance model for empirically testing new end-user information systems : theory and results*. Thesis, Massachusetts Institute of Technology.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 4171–4186.

Georgiana Dinu and Marco Baroni. 2014. How to make words with vectors: Phrase generation in distributional semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 624–633.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics* 9:391–409.

Shi Feng, Sidney D'Mello, and Arthur C Graesser. 2013. Mind wandering while reading easy and difficult texts. *Psychonomic bulletin & review* 20(3):586—592.

Lily Wong Fillmore and Charles J Fillmore. 2012. What does text complexity mean for english learners and language minority students. *Understanding language: Language, literacy, and learning in the content areas* pages 64–74.

Robert Flynn and Matthew Shardlow. 2021. Manchester metropolitan at SemEval-2021 task 1: Convolutional networks for complex word identification. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*. Association for Computational Linguistics, Online, pages 603–608.

Artyom Gadetsky, Ilya Yakubovskiy, and Dmitry Vetrov. 2018. Conditional generators of words definitions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Melbourne, Australia, pages 266–271.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna Clinciu, Dipanjan Das, Kaustubh D. Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Rubungo Andre Niyongabo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM Benchmark: Natural Language Generation, its Evaluation and Metrics. *arXiv e-prints* page arXiv:2102.01672.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence* 2(11):665–673.

Sian Gooding and Ekaterina Kochmar. 2018. 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*. Association for Computational Linguistics, New Orleans, Louisiana, pages 184–194.

Sian Gooding and Ekaterina Kochmar. 2019a. Complex word identification as a sequence labelling task. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, pages 1148–1153.

Sian Gooding and Ekaterina Kochmar. 2019b. Recursive Context-Aware Lexical Simplification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, pages 4852–4862.

Sian Gooding, Shiva Taslimipoor, and Ekaterina Kochmar. 2020. Incorporating multiword expressions in phrase complexity estimation. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*. European Language Resources Association, Marseille, France, pages 14–19.

John T. Guthrie, Allan Wigfield, Jamie L. Metsala, and Kathleen E. Cox. 1999. Motivational and cognitive predictors of text comprehension and reading amount. *Scientific Studies of Reading* 3(3):231–256.

L. Habib, G. Berget, F. E. Sandnes, N. Sanderson, P. Kahn, S. Fagernes, and A. Olcay. 2012. Dyslexic students in higher education and virtual learning environments: an exploratory study. *Journal of Computer Assisted Learning* 28(6):574–584.

M.A.K. Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. Routledge, 1st edition.

Arturo Hernandez and Ping li. 2007. Age of acquisition: Its neural and computational mechanisms. *Psychological bulletin* 133:638–50.

Arian Hosseini, Siva Reddy, Dzmitry Bahdanau, R Devon Hjelm, Alessandro Sordoni, and Aaron Courville. 2021. Understanding by understanding not: Modeling negation in language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, pages 1301–1312.

JISC. 2002. MLEs and VLEs explained. Accessed: 23-07-2021.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734* .

Michael L. Kamil, P. David Pearson, Elizabeth Birr Moje, and Peter Afflerbach, editors. 2010. *Handbook of Reading Research, Volume IV*. Routledge, 1st edition.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *CoRR* abs/2004.04906.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* .

Małgorzata Kuczera, Simon Field, and Hendrickje Catriona Windisch. 2016. Building skills for all: a review of england. *Policy Insights from the Survey of Adult Skills. OECD Skills Studies. Retrieved from https://www. oecd. org/unitedkingdom/building-skills-for-all-reviewof-england. pdf* .

V. Kuperman, H. Stadthagen-González, and M. Brysbaert. 2012. Age-of-acquisition ratings for 30,000 english words. *Behavior Research Methods* 44:978–990.

Nicole Landi, Stephen J. Frost, W. Einar Mencl, Rebecca Sandak†, and Kenneth R. Pugh. 2013. Neurobiological bases of reading comprehension: Insights from neuroimaging studies of word-level and text-level processing in skilled and impaired readers. *Reading & Writing Quarterly* 29(2):145–167.

Alberto Lavelli, Fabrizio Sebastiani, and Roberto Zanoli. 2004. Distributional term representations: An experimental comparison. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, New York, NY, USA, CIKM '04, page 615–624.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR* abs/1910.13461.

Jia Li, Chongyang Tao, Wei Wu, Yansong Feng, Dongyan Zhao, and Rui Yan. 2019. Sampling matters! an empirical study of negative sampling strategies for learning of matching models in retrieval-based dialogue systems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, pages 1291–1296.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, pages 74–81.

Diane Litman. 2016. Natural language processing for enhancing teaching and learning.

In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, AAAI'16, page 4170–4176.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, Dense, and Attentional Representations for Text Retrieval. *arXiv e-prints* page arXiv:2005.00181.

Christopher D. Manning and Hinrich Schutze. 1999. *Foundations of statistical natural language processing*. MIT Press, Cambridge, Mass.

Linda J. McCown. 2010. Blended courses: the best of online and traditional formats. *Clinical Laboratory Science: Journal of the American Society for Medical Technology* 23(4):205–211.

Ralph Meulenbroeks. 2020. Suddenly fully online: A case study of a blended university course moving online during the covid-19 pandemic. *Heliyon* 6:e05728.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM* 38(11):39–41.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2016. Definition modeling: Learning to define word embeddings in natural language. *CoRR* abs/1612.00394.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 2241–2252.

Evan Ortlieb, Stephan Sargent, and Meagan Moreland. 2014. Evaluating the efficacy of using a digital reading environment to improve reading comprehension within a reading clinic. *Reading Psychology* 35(5):397–421.

Gustavo Paetzold and Lucia Specia. 2015. LEXenstein: A framework for lexical simplification. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*. Association for Computational Linguistics and The Asian Federation of Natural Language Processing, Beijing, China, pages 85–90.

Gustavo Paetzold and Lucia Specia. 2016. 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*. The COLING 2016 Organizing Committee, Osaka, Japan, pages 1669–1679.

Gustavo H. Paetzold and Lucia Specia. 2017. A survey on lexical simplification. *J. Artif. Int. Res.* 60(1):549–593.

Alok Ranjan Pal and Diganta Saha. 2015. Word sense disambiguation: a survey. *CoRR* abs/1508.01346.

Chunguang Pan, Bingyan Song, Shengguang Wang, and Zhipeng Luo. 2021. DeepBlueAI at SemEval-2021 task 1: Lexical complexity prediction with a deep ensemble approach. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*. Association for Computational Linguistics, Online, pages 578–584.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pages 311–318.

Gustavo Penha and Claudia Hauff. 2021. On the calibration and uncertainty of neural learning to rank models. *CoRR* abs/2101.04356.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543.

Charles Perfetti and Suzanne Adlof. 2012. Reading comprehension: A conceptual framework from word meaning to text meaning. *Measuring Up: Advances in How We Assess Reading Ability* pages 3–20.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR* abs/1802.05365.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 1267–1273.

Maja Popović. 2018. Complex word identification using character n-grams. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, New Orleans, Louisiana, pages 341–348.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2020. Lsbert: A simple framework for lexical simplification. *CoRR* abs/2006.14939.

Mohammed Qrqez and Radzuwan Ab Rashid. 2017. Reading comprehension difficulties among efl learners: The case of first and second -year students at yarmouk university in jordan. *Arab World English Journal* 8:421–431.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering.

Maury Quijada and Julie Medero. 2016. HMC at SemEval-2016 task 11: Identifying complex words using depth-limited decision trees. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 1034–1037.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, Valencia, Spain, pages 99–110.

K Rayner and SA Duffy. 1986. Lexical complexity and fixation times in reading: effects of word frequency, verb complexity, and lexical ambiguity. *Memory & cognition* 14(3):191—201.

Paul Rayson, Scott Piao, Serge Sharoff, Stefan Evert, and Begoña Villada Moirón. 2010. Multiword expressions: hard going or plain sailing? *Language Resources and Evaluation* 44(1/2):1–5.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, pages 3982–3992.

Luz Rello, Ricardo Baeza-Yates, Stefan Bott, and Horacio Saggion. 2013a. *Simplify or Help? Text Simplification Strategies for People with Dyslexia*, Association for Computing Machinery, New York, NY, USA, pages 1–10.

Luz Rello, Ricardo Baeza-Yates, Laura Dempere-Marco, and Horacio Saggion. 2013b. Frequent words improve readability and short words improve understandability for people with dyslexia. In *Human-Computer Interaction – INTERACT 2013*. Springer Berlin Heidelberg, Berlin, Heidelberg, pages 203–219.

Irina Rets and Jekaterina Rogaten. 2021. To simplify or not? facilitating english l2 users' comprehension and processing of open educational resources in english using text simplification. *Journal of Computer Assisted Learning* 37(3):705–717.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword Expressions: A Pain in the Neck for NLP. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, and Alexander Gelbukh, editors, *Computational Linguistics and Intelligent Text Processing*, Springer Berlin Heidelberg, Berlin, Heidelberg, volume 2276, pages 1–15.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, pages 7881–7892.

Beth Sennett. 2016. Adjusting existing vles to support students with dyslexia. *Journal of Neurodiversity in Higher Education* 2:88–103. No DOI.

Donald Shankweiler, Eric Lundquist, Leonard Katz, Karla K. Stuebing, Jack M. Fletcher, Susan Brady, Anne Fowler, Lois G. Dreyer, Karen E. Marchione, Sally E. Shaywitz, and Bennett A. Shaywitz. 1999. Comprehension and decoding: Patterns of association in children with reading difficulties. *Scientific Studies of Reading* 3:69–94.

Matthew Shardlow. 2013. A comparison of techniques to automatically identify complex words. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*. Association for Computational Linguistics, Sofia, Bulgaria, pages 103–109.

Matthew Shardlow. 2014. Out in the open: Finding and categorising errors in the lexical simplification pipeline. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland, pages 1583–1590.

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. Complex - A new corpus for lexical complexity predicition from likert scale data. *CoRR* abs/2003.07008.

Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021a. Semeval-2021 task 1: Lexical complexity prediction. *CoRR* abs/2106.00473.

Matthew Shardlow, Sam Sellar, and David Rousell. 2021b. Collaborative augmentation and simplification of text (CoAST): pedagogical applications of natural language processing in digital learning environments. *Learning Environments Research* .

Kim Cheng Sheang. 2019. Multilingual complex word identification: Convolutional neural networks with morphological and linguistic features. In *Proceedings of the Student Research Workshop Associated with RANLP 2019*. INCOMA Ltd., Varna, Bulgaria, pages 83–89.

Vered Shwartz and Ido Dagan. 2019. Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics* 7:403–419.

Alexander Soemer and Ulrich Schiefele. 2019. Text difficulty, topic interest, and mind wandering during reading. *Learning and Instruction* 61:12–22.

Jared Stein. 2014. *Essentials for blended learning: a standards-based guide*. Essentials of online learning series. Routledge, New York.

Simeng Sun, Ori Shapira, Ido Dagan, and Ani Nenkova. 2019. How to compare summarizers without target length? pitfalls, solutions and re-examination of the neural summarization literature. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 21–29.

Aleksandra Szymańska and Alicja Wujec Kaczmarek. 2011. READING EFFICIENCY IN BLENDED LEARNING CONTEXT. *Teaching English with Technology* 11(2):29–42.

Walter van Heuven, Paweł Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2014. Subtlex-uk: a new and improved word frequency database for british english. *Quarterly journal of experimental psychology (2006)* 67.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR* abs/1706.03762.

Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria Aluísio. 2009. Facilita: reading assistance for low-literacy readers. In *Proceedings of the 27th ACM international conference on Design of communication - SIGDOC '09*. ACM Press, Bloomington, Indiana, USA, page 29.

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

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, Online, pages 2734–2744.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo H. Paetzold, Lucia Specia, Sanja Stajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. *CoRR* abs/1804.09132.

Seid Muhie Yimam, Sanja Štajner, Martin Riedl, and Chris Biemann. 2017. 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)*. Asian Federation of Natural Language Processing, Taipei, Taiwan, pages 401–407.

Marcos Zampieri, Liling Tan, and Josef van Genabith. 2016. MacSaar at SemEval-2016 task 11: Zipfian and character features for ComplexWord identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 1001–1005.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019a. Bertscore: Evaluating text generation with BERT. *CoRR* abs/1904.09675.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019b. PAWS: paraphrase adversaries from word scrambling. *CoRR* abs/1904.01130.

George Zipf. 1935. *The Psychobiology of Language: An Introduction to Dynamic Philology*. M.I.T. Press, Cambridge, Mass.

George K. Zipf. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley Press.

Manchester Metropolitan at SemEval-2021 Task 1: Convolutional Networks for Complex Word Identification

Robert Flynn
School of Computing, Mathematics
and Digital Technology
Manchester Metropolitan University
robert.flynn@stu.mmu.ac.uk

Matthew Shardlow
School of Computing, Mathematics
and Digital Technology
Manchester Metropolitan University
m.shardlow@mmu.ac.uk

Abstract

We present two convolutional neural networks for predicting the complexity of words and phrases in context on a continuous scale. Both models utilize word and character embeddings alongside lexical features as inputs. Our system displays reasonable results with a Pearson correlation of 0.7754 on the task as a whole. We highlight the limitations of this method in properly assessing the context of the target text, and explore the effectiveness of both systems across a range of genres. Both models were submitted as part of LCP 2021, which focuses on the identification of complex words and phrases as a context dependent, regression based task.

1 Introduction

Complex Word Identification (CWI) involves identifying words that the reader may find difficult to understand. A word's complexity can depend on many factors and differ according to context. Further, assessment of the complexity of named entities can require some degree of general knowledge, making CWI a challenging task (Shardlow, 2013). Accurately identifying complex words is important for many downstream simplification tasks, making literature more accessible for people with conditions such as dyslexia (Rello et al., 2013), and the assessment of a text's readability as a whole (Dubay, 2004).

Our methodology plans to extend on previous convolutional network based approaches to CWI (Aroyehun et al., 2018; Sheang, 2019). With the goal of producing a system that is able to better model the complexities of phrases and unfamiliar words, within the English language.

Previous shared tasks on CWI addressed the problem as a binary and probabilistic classification type task, although human judgements on word complexity are not binary and exist on a continuous scale. Lexical Complexity Prediction (LCP) 2021 tries to address this and uses an augmented version of CompLex (Shardlow et al., 2020), a dataset annotated with a 5-point Likert scale. CompLex also features context-specific annotation, with words receiving different annotations depending on their context. The dataset provides annotations from three different domains: Bible, Biomed and Europarl (Shardlow et al., 2021).

The code for this task is available on GitHub[1].

2 Related Work

Word frequency is a commonly used feature in CWI (Gooding and Kochmar, 2018; Kajiwara and Komachi, 2018); words that appear frequently in language are more likely to be recognised and understood by the reader (Carroll et al., 1998). For the purpose of identifying medical terminology that may be unfamiliar to the lay reader, Elhadad (2006) leveraged lexical frequencies while also exploring the potential of other features such as word familiarity ratings from the MRC Psycholinguistic Database (Coltheart, 1981).

More recently lexical and psycholinguistic features have been utilized by machine learning tools, resulting in improved accuracy on these tasks. Through the use of an enseble-based voting method the CAMB system (Gooding and Kochmar, 2018) achieved state-of-the-art results in the 2018 CWI shared task (Yimam et al., 2018), employing a total of 27 lexical, morphological and psycholinguistic features. The CAMB system however does not consider the target words context, opting for a "greedy" approach towards phrase classification, marking all phrases as complex.

Aroyehun et al. (2018) explored the use of convolutional neural networks (CNN) for CWI using only

[1] https://github.com/robflynnyh/CNN-LCP-Shared-Task-2021

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 603–608
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

the word embeddings of the target words and the averaged embeddings of the left and right contexts as inputs. They contrasted the results against a feature engineering approach using decision tree learning finding that both methods achieved competitive results. However, their decision tree method was marginally more accurate than their CNN for most of the datasets. Integrating lexical features alongside word embeddings can lead to further improvements in accuracy making this a more competitive approach, and outperforming many previous deep learning methods for CWI (Sheang, 2019).

By framing CWI as a sequence labelling task, Bi-directional long short-term memory (BiLSTM) networks have produced state-of-the-art results on the CWIG3G2 dataset (Yimam et al., 2017; Gooding and Kochmar, 2019). BiLSTM networks are able to capture long-term word and character level dependencies allowing these models to consider a large amount of contextual information. Modelling the complexity of phrases has proven to be a more challenging and complex task compared to individual words (Gooding and Kochmar, 2019).

3 Implementation

3.1 Features

Below a description of the features used by both models is given:

Frequency: Word frequencies are taken from the SUBTLEX-UK word frequency database (van Heuven et al., 2014). Logarithmic Zipf frequency values were chosen based on previous results from this metric (Zampieri et al., 2016) and the Zipfian distribution that is displayed in language (Zipf, 1949).

Age of Acquisition: Age of Acquisition (AoA) values, estimating the age at which a word is typically acquired. (Kuperman et al., 2012; Brysbaert, 2012).

Word-level Features: Target word length and number of syllables are used as features (Brysbaert, 2012).

Corpus Type: As the dataset includes extracts from three different sources of potentially varying complexity, the corpus type was included and represented as a one-hot embedding.

Pre-trained Embeddings: 50d GloVe (Pennington et al., 2014) word embeddings, and 50d chars2vec[2] embeddings representing a word's character sequence are used. 50d GloVe embeddings were chosen as embeddings with more dimensions showed worse performance on the training data. Which suggests that the 50d embeddings capture sufficient information needed for this task. Character embeddings allow inferences to be made between words with similar morphologies.

3.2 Preprocessing

Firstly min-max normalization is applied to the features taken from datasets, and word length is divided by 10. Non-alphanumeric characters are removed from the sentences before any features are extracted.

Both models take as inputs the features for the target word, and the averaged features for the left and right contexts of the target text. If the target word or words are positioned at the beginning or end of the sentence a zero vector of size 107 is used for the left or right context. For out-of-vocabulary words a zero vector is used for the word embedding and other features are imputed using mean values from their respective datasets. Finally the vectors for the target text and its context are stacked to produce a 3x107 matrix (left context — token — right context) for single words or a 4x107 matrix for MWEs (left context — token 1 — token 2 — right context).

3.3 Models

This section provides a description of the architecture and hyperparameters used for both models. The models were produced using the Keras library version 2.4.3. Each of the models were trained with a batch size of 50, early stopping of 1000 and model checkpointing based on the validation loss.

3.3.1 Single Word Model

For single words a 1D convolutional network followed by three fully connected layers is used. The model takes three inputs, an average of the features for left and right contexts is used for the first and third inputs respectively, and the features of the target word is used as the second input. The convolutional layer pads the inputs and uses a kernel size of 3 with 150 output filters and ReLu as the activation function. Global Max Pooling and a flatten layer followed by batch normalization is then applied to the output of this layer. Three dense layers with sizes of 150 (ReLu), 50 (ReLu) and 1 (Linear) are then used with a Dropout of 0.5 applied before each dense layer. Mean Squared Error

[2] https://github.com/IntuitionEngineeringTeam/chars2vec

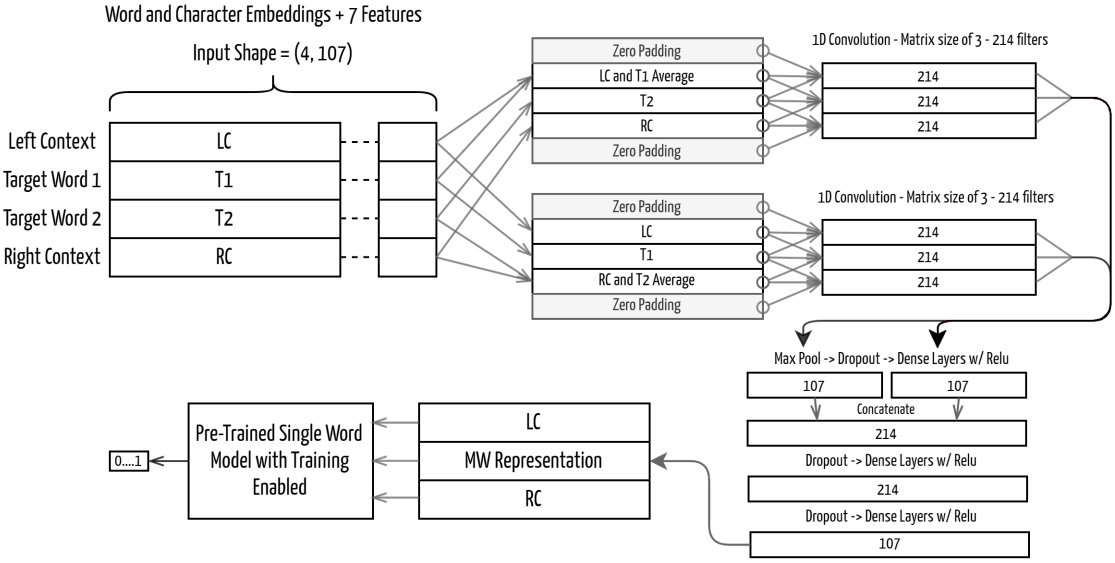

Word and Character Embeddings + 7 Features
Input Shape = (4, 107)
Left Context
Target Word 1
Target Word 2
Right Context
LC
T1
T2
RC
Zero Padding
LC and T1 Average
T2
RC
Zero Padding
Zero Padding
LC
T1
RC and T2 Average
Zero Padding
1D Convolution - Matrix size of 3 - 214 filters
214
214
214
1D Convolution - Matrix size of 3 - 214 filters
214
214
214
Max Pool -> Dropout -> Dense Layers w/ Relu
107
107
Concatenate
214
Dropout -> Dense Layers w/ Relu
214
Dropout -> Dense Layers w/ Relu
107
LC
MW Representation
RC
Pre-Trained Single Word Model with Training Enabled
0....1

Figure 1: Depiction of multi-word model architecture

(MSE) is used as the loss function and Stochastic Gradient Descent as the optimizer, with a learning rate of 0.01 and momentum of 0.6 with Nesterov accelerated gradient enabled.

3.3.2 Multi-Word Model

For multi-words a second model is used to assess the complexity of two word phrases. This model acts as an adapter with the output being fed into a pre-trained single word model, allowing the model to take advantage of the data for single words and MWEs. Figure 1 gives an overview of the model architecture.

Features for the averaged left context, target word one, target word two and the averaged right context are used as input for the model. A convolutional layer with a similar architecture to task one is used for each of the target words. For the two convolutional layers the other target word is averaged with either the left or right context depending on its positioning, weighting the other target word higher than the rest of the context.

Each convolutional layer uses a filter size of 214 but is otherwise the same as in task one. Global Max Pooling followed by Dropouts of 0.3 and dense layers with 107 neurons and ReLu activation are applied to the outputs of the convolutions which are then concatenated along the last axis. Two dense layers with ReLu activation and sizes of 214 and 107 are then applied with a Dropout of 0.5 before each layer. This final output of size 107 is then concatenated along the first axis with the original left and right contexts to form the input for a pre-trained single word model with training enabled. This model uses the Adam optimizer with default parameters and MSE as the loss function.

4 Results

Task	Pearson	MSE	R2
Task 1	0.7389	0.0074	0.5398
Task 2	0.7754	0.0079	0.5989

Table 1: Results for both tasks

This section will discuss and evaluate the performance of both models. Participants were ranked according to the Pearson correlation coefficient of their submissions. Table 1 presents the results for each of the tasks with task 1 evaluating individual words and task 2 covering both single and two word Multi-Word Expressions (MWEs).

4.1 Single Word Model Results

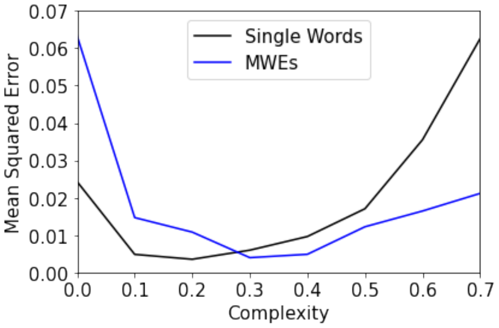


Figure 2: MSE across different complexities

As shown in Figure 2 the single word model struggles to accurately predict values for words of a high complexity, and also displays difficulties for words of a complexity of less than 0.1. The training and evaluation data features less examples of very simple or complex words. The complexity of these extremities is often highly dependant on the context, making them more challenging to assess.

Corpus	Pearson	MSE	R2
All	0.7389	0.0074	0.5398
Bible	0.7085	0.0085	0.4948
Biomed	0.7828	0.0087	0.6050
Europarl	0.6807	0.0055	0.4562
JUST-BLUE	0.7886	0.0062	0.6172

Table 2: Results for individual words

Table 2 presents the results for this task on each of the domains and the task as a whole. The prediction accuracy varies significantly across the different sources. Results from the best performing team are given for comparison (Shardlow et al., 2021).

As the model only uses an average of the features present in the left and right context of the target word, it is unable to differentiate between tokens that are influential to the target words complexity and ones that are not. Because of this equal weighting of words in the context, the models accuracy can be negatively affected by an abundance or lack of stop words in the sentence. Very complicated or simple words in sentence that are not related to the target word, and don't share a similar complexity can also cause the model to over- or under-predict the target word's complexity. The mechanism by which the model assesses the context may partly explain the variance in accuracy on each domain.

Interestingly, our sub-analysis showed that the model shows a better correlation for those tokens without a word embedding, yielding a Pearson correlation of 0.7804 and a MSE of 0.0071. Generally these out-of-vocabulary words are more complex so the model is using the lack of a word embedding as a feature when making predictions. Although this shows a better correlation overall it could lead to false positives in specific cases where the out-of-vocabulary word is of a low complexity.

4.2 Multi-Word Model Results

As shown in Figure 2 the multi-word model is much less accurate for very simple MWEs of a complexity less than 0.1. However, for more complex words the predictions remain fairly accurate. This model is able to asses the way in which the words in a phrase interact with each other and to some degree the rest of the sentence. This additional contextual information may increase the model's capacity to evaluate more complex words. Only 1.65 percent of phrases in the training data were of a complexity of less than or equal to 0.1 which could explain the inaccuracy in this range.

Corpus	Pearson	MSE	R2
All	0.7611	0.0102	0.5770
Bible	0.7173	0.0113	0.5106
Biomed	0.7980	0.0141	0.6317
Europarl	0.5799	0.0060	0.3089
DeepBlueAI	0.8612	0.0063	0.7389

Table 3: Results for MWEs

MWE Type	Pearson	MSE	R2
A-N (115)	0.7654	0.0115	0.5801
N-N (56)	0.7414	0.0091	0.5293

Table 4: Results for the different MWE formations. A-N: Adjective-Noun. N-N: Noun-Noun.

Table 3 presents the results across each of the different domains present in the dataset. The model used for MWEs makes use of a fine-tuned instance of the single-word model; consequentially incorrect associations from the single-word model may have been carried over to this model. The results show a similar variance across domains to task 1, although it struggles more significantly on the Europarl sub-corpus. Compared to the other domains, Europarl's complexity values show a much smaller standard deviation than the other sub-corpora (0.093 compared to 0.196 and 0.152, on biomed and bible). The variation of complexities may play a role in the models effectiveness at making accurate predictions across the domains.

Table 4 presents the results across different MWE formations. The number of occurrences of each part-of-speech formation is denoted in brackets, MWE types with less than 10 occurrences were omitted from the table. The model performs marginally better on Adjective-Noun MWE formations.

5 Discussion

In this paper, we presented two convolutional neural networks used as an approach to single-word and multi-word complex word identification. Both models achieved reasonable results, achieving scores in a comparable range to the majority of other submissions.

Multi-Word CWI is a more challenging task compared to the assessment of single words; the multi-word model was able to utilize the datasets of both tasks, and its predictions show a Pearson's correlation score of 0.7611. Our system is only able to process two-word MWEs, which for this task is not an issue. However, in other use cases the ability to assess longer MWEs would be useful. Given a dataset with annotations for longer MWEs the model could potentially be adapted to work with three or four word sequences.

Both models are able to assess the context of the target text when making predictions; although, as the left and right contexts are given as an average, all words are weighted equally regardless of their relevance to the target text. Because of this equal weighting of words, the models are able to adjust their predictions based on the general complexity of the sentence but are unable to fully capture the relevant context. Adding a mechanism that could weight each word in the context based on certain features may offer some improvements in this area. Attention based models such as BERT (Devlin et al., 2019) are able to attend to each token in a sequence to produce embeddings that capture large amounts of contextual information. Fine-tuning such a model on CWI tasks could produce embeddings that contain more useful and relevant information.

References

Segun Taofeek Aroyehun, Jason Angel, Daniel Alejandro Pérez Alvarez, and Alexander Gelbukh. 2018. Complex word identification: Convolutional neural network vs. feature engineering. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 322–327, New Orleans, Louisiana. Association for Computational Linguistics.

Marc Brysbaert. 2012. crr " age-of-acquisition (aoa) norms for over 50 thousand english words.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. *Proc. of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William Dubay. 2004. The principles of readability. *CA*, 92627949:631–3309.

Noemie Elhadad. 2006. Comprehending technical texts: predicting and defining unfamiliar terms. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, 2006:239–243. 17238339[pmid].

Sian Gooding and Ekaterina Kochmar. 2018. 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. Association for Computational Linguistics.

Sian Gooding and Ekaterina Kochmar. 2019. Complex word identification as a sequence labelling task. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1148–1153, Florence, Italy. Association for Computational Linguistics.

Walter van Heuven, Paweł Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2014. Subtlex-uk: a new and improved word frequency database for british english. *Quarterly journal of experimental psychology (2006)*, 67.

Tomoyuki Kajiwara and Mamoru Komachi. 2018. 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. Association for Computational Linguistics.

Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. 2012. Age-of-acquisition ratings for 30,000 english words. *Behavior Research Methods*, 44(4):978–990.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Luz Rello, Ricardo Baeza-Yates, Stefan Bott, and Horacio Saggion. 2013. Simplify or help? text simplification strategies for people with dyslexia. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, W4A '13, New York, NY, USA. Association for Computing Machinery.

Matthew Shardlow. 2013. A comparison of techniques to automatically identify complex words. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, pages 103–109, Sofia, Bulgaria. Association for Computational Linguistics.

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. CompLex — a new corpus for lexical complexity prediction from Likert Scale data. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*, pages 57–62, Marseille, France. European Language Resources Association.

Matthew Shardlow, Richard Evans, and Marcos Zampieri. 2021. Predicting lexical complexity in english texts. *arXiv preprint arXiv:2102.08773*.

Kim Cheng Sheang. 2019. Multilingual complex word identification: Convolutional neural networks with morphological and linguistic features. In *Proceedings of the Student Research Workshop Associated with RANLP 2019*, pages 83–89, Varna, Bulgaria. INCOMA Ltd.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 66–78, New Orleans, Louisiana. Association for Computational Linguistics.

Seid Muhie Yimam, Sanja Štajner, Martin Riedl, and Chris Biemann. 2017. 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. Asian Federation of Natural Language Processing.

Marcos Zampieri, Liling Tan, and Josef van Genabith. 2016. MacSaar at SemEval-2016 task 11: Zipfian and character features for ComplexWord identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1001–1005, San Diego, California. Association for Computational Linguistics.

George K. Zipf. 1949. *Human Behavior and the Principle of Least Effort*.