# Generating Vocabulary Sets for Implicit Language Learning using Masked Language Modeling

Vatricia Edgar
School of Computing and Augmented Intelligence (SCAI)
Arizona State University
vcedgar@asu.edu

Ajay Bansal
School of Computing and Augmented Intelligence (SCAI)
Arizona State University
ajay.bansal@asu.edu

## Abstract

*A well-balanced language curriculum must include both explicit vocabulary learning and implicit vocabulary learning. However, most language learning applications focus on explicit instruction. Students require support with implicit vocabulary learning because they need enough context to guess and acquire new words. Traditional techniques aim to teach students enough vocabulary to comprehend the text, thus enabling them to acquire new words. Despite the wide variety of support for vocabulary learning offered by learning applications today, few offer guidance on how to select an optimal vocabulary study set. This paper proposes a novel method of student modeling with masked language modeling to detect words that are required for comprehension of a text. It explores the efficacy of using deep learning via a pre-trained masked language model to model human reading comprehension and presents a vocabulary study set generation pipeline (VSGP). Promising results show that masked language modeling can be used to model human comprehension and the pipeline produces reasonably sized vocabulary study sets that can be integrated into language learning systems.*

## 1. Introduction

Language learners must acquire a sizable vocabulary. An optimal understanding of texts in the language occurs with 8,000 word-families known [1]. Digital language learning platforms address this need for vocabulary with wide array of vocabulary training options. For example, studies found that most popular mobile language learning applications focus solely on vocabulary [2]. However, most of the applications surveyed opt to teach via behavioral memorization techniques over communitive, message-focused learning [2], despite research showing that implicit, message-focused learning is an important part of a balanced language curriculum [3].

Implicit learning, is defined by as "the incidental, as opposed to intentional, derivation and learning of new word meanings by subjects reading under reading circumstances that are familiar to them [4]." Implicit learning requires support because to learn effectively, students must read texts that they can comprehend. A major factor in whether unknown words in a text can be guessed and acquired is the number of words in the text that are already known to the student. A meta-analysis of incidental vocabulary research showed that 15% of unknown words are learned when 97% of words or more are already known [4]. This research is in accordance with research by [1] which shows that language learners must know between 95% and 98% of the vocabulary in a text to comprehend it the same amount as native speakers.

Two techniques that have been developed to help students learn vocabulary incidentally are vocabulary pre-teaching and glossing. Vocabulary pre-teaching teaches the words required for comprehension before reading [3]. Instructors and publishers preparing materials for pre-teaching must make assumptions about what words students do not know, and which of those will cause comprehension difficulties. Therefore, there is a lack of personalization, as some words that should be pre-taught for one student might be easy for another. Glosses include the information about unknown words inside the text. The author of a glossed text also cannot personalize glosses for all students. One solution provided by some digital glossing programs is to have glosses for any word that students want [5]. However, it is unclear if unlimited look up of words is optimal. It has been shown that up to a third of language learners may use word lookups excessively [3], which is disruptive and a waste of learning time [6].

We propose an open-source pipeline for generating student-specific and text-specific vocabulary study sets for use either with vocabulary pre-teaching or with limited glossing. This tool aims to create personalized vocabulary study sets specific to the structure of the text being read and to the vocabulary knowledge of the student studying the text, without requiring large amounts of data about the student or similar students. A vocabulary set is produced by modeling the students' ability to guess the unknown words in a text. The set of all words that are impossible or difficult to guess make up the study set. The student is modeled by a language model that uses deep learning, to predict words that are unknown. The framework is designed to be flexible, allowing it to be easily integrated into new language learning systems or can be used to build new information systems software.

This work is organized as follows: in section II, we discuss works related to automatic selection of vocabulary for study. Next, we present our proposed approach of using language modeling as student modeling, followed by a

URI: https://hdl.handle.net/10125/79425
978-0-9981331-5-7

discussion of the design of the vocabulary set generation pipeline in Section IV. Then we describe of the methodology for testing the models used by the pipeline in section V followed by a discussion of examining the created example study sets in section VI. The results of both the model testing and of the sample sets are presented in section VII, and finally, present conclusions and impact.

## 2. Related Work

We discovered just one popular language learning application which offers automatic selection of vocabulary study sets: Vocabulary.com. The automatically generated study sets can be studied on the website through gamified methods [7]. Vocabulary.com is a popular tool and instructors have reported that it increases vocabulary knowledge, reading comprehension, and student engagement. However, the website attributes the source of their success to the millions of responses they have collected for over 100,000 questions. Unfortunately, the use of large data sets of student questions and answers prohibits new applications from initially using a method such as this, since a new application will not have collected this much data

To understand what vocabulary should be in a vocabulary study set, we analyzed vocabulary glossing and pre-teaching studies to examine the process of selecting vocabulary sets for these methods. Two studies stood out [8], [9]. These used a pilot study to determine which words in the target text would cause difficulty for students of the main study. These studies asked students to identify unknown words in the text used for the study, and to attempt to guess them as well, to control for context. Words unknown and unguessable for 60% or more students in the pilot study were selected for glossing. This inspired the proposed approach in this paper, i.e., application attempts to predict words that cannot be guessed by student.

To support the goal of automatically determining what words can be guessed, the automatic prediction of word difficulty in a text was studied. Cloze (fill-in-the-blank) tests are language tests which assess students' guessing-in-context ability, among other skills. When automatically generating or evaluating cloze tests, the "guessability" of target words (the words left blank to be guessed) must be carefully considered. Target words must have enough context in the surrounding text to allow the student to correctly guess them. We explore three papers which focus on the selection or evaluation of target words in cloze tests.

A context-based method of generating multiple choice cloze tests based on input text given by the user was proposed [10]. The paper makes a distinction between the full context of a complete sentence or paragraph and the narrow context of a smaller group of words, two to five words in length. According to the paper, tests should have target words which are obviously correct given the full context of the text, and distractors which are equally plausible given a narrow context. Word co-occurrence is used to determine which words have the most context available in the full scope of the text, because words that occur frequently together are likely to provide context for each other. Google N-gram is used to find suitable distractors. Using n-grams made of the words surrounding the target word, with the target replaced by possible distractors, the distractors with the most common n-grams are chosen, as these words appear in the local context most frequently. This method resulted in an average of more than 90% of target words fitting the full and narrow contexts, meaning most target words should be easily guessable in both contexts by students with good reading comprehension. The method had the best results in choosing words that fit the narrow context when using larger n-grams, with an accuracy of about 74%.

A method of evaluating open cloze tests using entropy to model the restrictiveness of the context provided around the target word was proposed [11]. The context surrounding target words in open cloze tests requires a limited context to reduce choices so students do not respond with unexpected answers that are technically correct. This study uses the number of possible syntactically and semantically correct choices for a gap and the probability of those choices to measure the restrictiveness of the context provided by questions. The number of choices and their probability is modeled with entropy, "which quantifies the amount of information conveyed by an event" [11]. 5-gram bi-directional language model was used to measure entropy. Questions with more options for responses and higher probability for those responses will have higher entropy. They measured the entropy of open cloze tests from Cambridge English examinations and found that entropy generally correlates to difficulty level.

The final paper on cloze test evaluation [12], predicts the difficulty of closed cloze tests as well as two other similar tests. The researchers predicted test difficulty with classification and regression models, originally introduced in [13]. They extract features in three ways. The first set of features, a super set of which was initially used in [12], includes properties of the solution, properties of the text, and properties of the question. Solution properties describe the target word and its immediate context (the words before and after the gap). Text properties refer to properties of the entire sentence or paragraph of the question and include readability measures. Test properties refer to properties of the test itself. Additional features introduced in [13] are the ability for language modeling and semantic relatedness to predict the correct answer. The language modelling method uses a 5-gram statistical language model to predict answers by calculating all possible answers and selecting the most probable resulting sentence. Semantic relatedness was used to account for long distance context. The similarity of each of the multiple-choice responses to each of the content words in the question sentence is calculated by finding the

cosine similarity between each word's word vector. Candidates that are most similar to other words are considered the most likely fit and are selected as the answer. Researchers found that the use of language modelling and semantic relatedness ability to predict words as features significantly improved the regression model's ability to predict text difficulty over the features derived from other properties of the question [13]. This suggests that ability for the language model or semantic relatedness methods to answer questions correctly does correlate with students' ability to answer questions correctly.

## 3. Proposed Approach: Language Modeling as Student Modeling

The use of language modeling in the related works is especially interesting. Previous studies [10], did not use actual language modeling but did use a method similar to n-gram statistical language modeling, as counting the number of times an n-gram appears in Google N-grams does approximate the probability of that n-gram. [11] use language modeling to measure both candidate space and candidate probability for use in calculating entropy, which models the quality of the context provided by the text. Study by [13] uses language modeling directly to predict difficulty of a text: when a language could not predict a word, that correlated with a higher percentage of students being unable to predict the word. This correlation shows that language models have the potential to model not only available context but also student guessing ability. They also used semantic relatedness to account for the language model's inability to deal with important context that is larger than 5-grams. Some advanced language modeling techniques can account for more context, which makes up for this deficiency.

It is not surprising that language models are used to model context and complexity in these work as language models encapsulate both semantic and syntactic information about a text. Language models have the advantage of removing the need for manual encoding of linguistic features, which makes it much easier to adapt applications using language models to many languages. For these reasons, we chose language modeling as the tool to determine the difficulty of unknown words in a text.

There are several forms of language models, including n-gram models, recurrent neural network models (RNN), long short-term memory RNN models (LSTM), and transformer models. To choose the proper type of model, we examined what we know about student reading ability to discover a model which may predict words in a similar fashion as humans. The first issue is that of long-distance context, as noted by [13], who needed to use semantic relatedness to model long distance connections. The second issue is of bidirectionality. When predicting words in context, students can and often must use words that come after the unknown word. The final issue pertains to the development of future digital learning applications. These language models should be easy to train without an inaccessible amount of data that explodes training time to unreasonable lengths.

When examining common language models, we can quickly reduce the list of possible models by removing n-gram models and RNNs. The n-gram is of fixed length and cannot expand to fit the length of whole sentences [14]. RNNs allow variable size input but suffer from the vanishing gradient problem and in practice do not consider much context [14]. LSTMs and transformers, on the other hand, are both more capable of handling long distance context, although transformers perform better [15], [14]. Both model types have bi-directional variants, however, bi-directional LSTMs considers the left and right contexts sequentially [14], while bidirectional transformers consider both in parallel (masked language modeling) [15]. While students read sequentially, seeing the left context first, then the right context, they can also jump around in the text after reading it. Students can make observations of context clues that occur before and after the word at any time and make connections between the left and right context. Parallel bidirectionality may model this better than sequential bidirectionality.

Both LSTM-based and transformer-based models are available under open licenses as pre-trained models, meaning that one only needs to download the models and libraries to use them [15], [16]. Both models were intended to be fine-tuned on specific language tasks, however, since they are language models and can be used for word prediction as is, this is not strictly necessary. We investigate the use of bi-directional transformers since we believe that the ability to process information in parallel is an important factor. In particular, we chose BERT, the original masked language model [15], to model students' ability to guess words in context.

## 4. Design of Vocabulary Set Generation

Here, we discuss the generation of vocabulary sets, given a text and the student's vocabulary. First, we describe the model used to represent the student, then the model used to represent the document, and finally, the complete pipeline together.

### 4.1 The Student Model

The Student Model contains information needed to discover unknown words and unguessable words in a text. The most important of these include a list of vocabulary known by the student, which can be gathered with a vocabulary estimation test (not presented in this paper), and configuration parameters specific to the student being modelled. The configuration parameters determine what BERT model (Google Research, 2020) to use and how to use it. The description of the model configurations, how they were selected and the results are described under

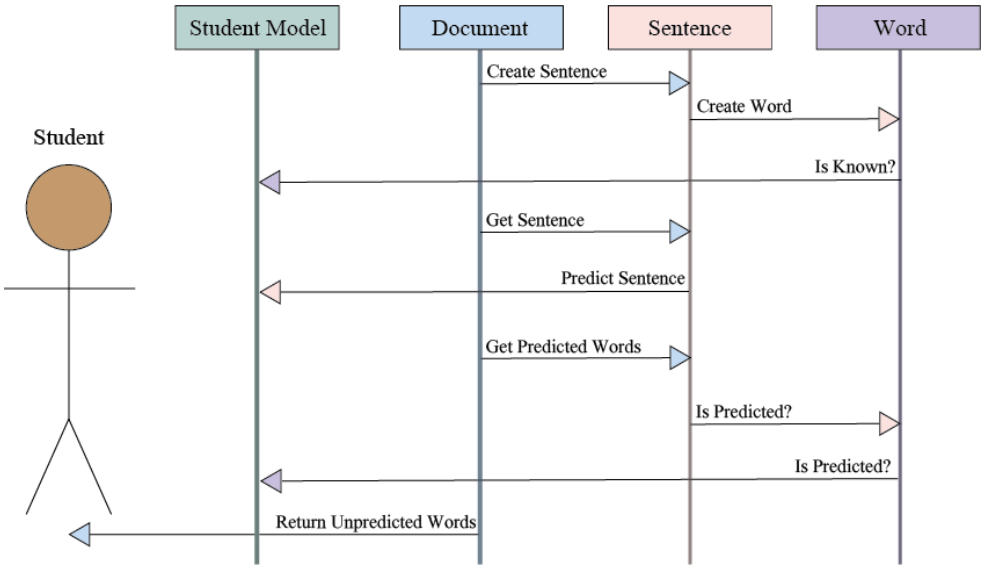


**Figure 1: The Vocabulary Set Generation Pipeline**

Methodology and Results. With this information, the Student Model performs three major functions. First, it determines if the student knows a word by determining if the word is a content word on the known word list for the student. Second, it predicts words in sentences to determine which words will be difficult for the student to guess. Last, it determines if a word has been predicted according to the configuration.

### 4.2 The Document Model

The Document Model consists of Sentences and Words which represent the text that the student is trying to read. The Document model also contains the Student Model to simulate reading of the text. The most fundamental part of the Document model is the Word class. The Word class includes the original tokens of the word (after being parsed by BERT's tokenizer) and the "model tokens" of the word. The model tokens are a list of the tokens that will be passed to the model for prediction. These tokens are equal to BERT's special masking token if the word is not known to the student, and equal to the actual tokens otherwise. This signals to BERT that the unknown words must be predicted. Words are the basic building blocks of the Sentence class. Like the Word class, the Sentence class contains the actual and model tokens of the sentence. To be processed by BERT, the model tokens list includes special start and end tokens, as well as padding tokens that fill the list to its maximum size. The Document class is made of Sentences. To create the Sentences, the Document uses spaCy[1] to extract each sentence from the text. Once a text is parsed into a Document, it is ready to be predicted.

### 4.3 The Complete Pipeline

The process of generating a vocabulary set is presented in Figure 1. After the Student Model is created, the Document model is created, which is shown in the yellow portion of Figure 1. When the Document creates the Sentences and Words that it is made of, the Student Model is used to determine which words should be masked. When the Document is predicted, as seen in orange in Figure 1. Then the Document Model cycles through the Sentences and uses the Student Model to the predict the sentence's tokens. Finally, each Word in each Sentence is updated, using the Student Model to determine if the word has been correctly predicted. This is in red in Figure 1. When all words have been marked as predicted or not predicted, the Document Model gathers the words that were not predicted. These are returned as the vocabulary study set.

## 5. Methodology

The methodology consists of two tests. First, we describe the method for comparing the model's guessing ability to human's guessing ability. We chose model configurations we believe will perform well based on a configuration data set and test the chosen configurations on a testing set. Second, in absence of a user study, we examine the output of the Vocabulary Set Generation Pipeline (VSGP) with test students and texts.

### 5.1 Configuring and Validating the Model

With cloze tests, we can have the model "answer" every question, then compare the model's ability to predict words to human's ability to predict words, as described below under "Procedures." Existing data sets were used. The

[1] Natural language processing in python (https://spacy:io/)

**Table 1: Overview of the Cloze Question Dataset**

| | | 25th Percentile | 50th Percentile | 60th Percentile |
|---|---|---|---|---|
| Beginner | Percent Easy Questions | 0.51 | 0.72 | 0.72 |
| | Percent Difficult Questions | 0.49 | 0.28 | 0.28 |
| | Total # of Easy Questions | 87 | 123 | 123 |
| | Total # of Difficult Questions | 83 | 47 | 47 |
| | | 25th Percentile | 50th Percentile | 60th Percentile |
| Intermediate | Percent Easy Questions | 0.53 | 0.78 | 0.82 |
| | Percent Difficult Questions | 0.48 | 0.23 | 0.18 |
| | Total # of Easy Questions | 105 | 155 | 164 |
| | Total # of Difficult Questions | 95 | 45 | 36 |
| | | 25th Percentile | 50th Percentile | 60th Percentile |
| Advanced | Percent Easy Questions | 0.66 | 0.89 | 0.92 |
| | Percent Difficult Questions | 0.34 | 0.11 | 0.08 |
| | Total # of Easy Questions | 132 | 178 | 184 |
| | Total # of Difficult Questions | 68 | 22 | 16 |
| | | 25th Percentile | 50th Percentile | 60th Percentile |
| All | Percent Easy Questions | 0.59 | 0.84 | 0.88 |
| | Percent Difficult Questions | 0.42 | 0.17 | 0.12 |
| | Total # of Easy Questions | 117 | 167 | 176 |
| | Total # of Difficult Questions | 83 | 33 | 24 |

original multiple-choice cloze questions are by [17]. The sentences that make up the cloze questions were selected from 5 Sherlock Holmes books. Each sentence contains a low frequency word which was selected as the that is left blank in the cloze question. The distractors are unimportant for the purpose of testing the models because the ultimate application of the model is to generate a correct choice.

The human responses to the cloze questions are made available by [18] and these responses were used by [19] and [12], whose study was already described in Related Work. Study [12] conducted a series of surveys to collect data about the difficulty for humans of the 200 cloze questions. The results of their study showed that there is a variety of both easy and difficult questions because there is a high standard deviation of the error rates (number of incorrect responses/total number of responses) for the questions in this set.

The surveys conducted were in the following form: Each survey contained 10 questions and asked students for information, including their self-identified CEFR (Common European Framework of Reference for Languages) level (A1, A2, B1, B2, C1, or C2). The students are then asked each of the ten multiple choice questions. According to [12] the students could take multiple surveys, but data was not collected to connect responses from each survey to specific students. Thus, we will approximate the number of students that the dataset contains by assuming that each student took one ten-question survey, so every ten responses represent a single student. Table 1 describes the data in detail. For the purpose of this study, students were grouped by their CEFR level into larger levels of advanced (C levels), intermediate (B levels), and beginner students (A levels). Related study [12] considered the students' reported CEFR levels were unreliable. We believe that this makes the data more realistic given that the VSGP currently relies on self-reported levels of beginner, intermediate, and advanced. The model's performance with the set including of all students was also analyzed.

Two types of preparation were made before the data could be used in the validation procedures. First the student levels are split further into percentiles roughly representing better or worse performing students who are at the same level. This grouping determines what questions in the data set are considered hard. If we are considering lower performing advanced students, than the questions where 25% or more advanced students answered incorrectly are considered difficult, whereas with high performing advanced students, threshold is raised to 50%. An additional grouping was also used, based on the method [8] used to determine which words from the pilot study should be glossed. In [8], words marked as unknown and unguessable by more than 60% of the students in the pilot study were chosen to be glossed. While we believe that the 25% and 50% levels provide opportunity for more customization, we also want to compare our work to the threshold determined by an expert in the domain.

The second type of preparation splits the data into configuring and testing data sets. The configuring set is used to determine the correct set of parameters for using the model with each student group. To validate the chosen parameters, we test them with a subset of the data. We opted for a split of 50% for configuration and 50% for testing. In order to overcome issues of class imbalance, where there is a possibility of having an extremely small amount of difficult words in one set and a large amount in

the other, we took the approach suggested by [20]. According to [20], test sets can be generated with 3 levels of granularity. The first level is entirely non-granular: the data is split randomly among the test and training sets. In the second level, which attempts to address class imbalance, the data set is first split into classes and the classes are then split into the training and test data sets. The training and test data sets for each class are then combined to create the training and test data sets for the entire set. The third level splits the data set further into subclasses and again partitions the data in these subclasses into training and test sets. For the purposes of our study, only the second level was considered to ensure that the model is configured and tested with a balanced set of data. This also ensures the distribution of easy and hard words for each student group in the original, configuring, and testing data sets. The data is split in the difficult and easy classes and 50% of each class is placed in the configuring set while the other 50% is placed in the testing set.

### 5.2 Procedures

As previously discussed under the pipeline, there are several parameters that must be examined to determine what works best for which level of student. The following parameters are considered in this study: the specific BERT model used, the number of predictions the model is allowed, the use of synonyms provided by the Meriam-Webster API, and the use of similarity scores provided by the Spacy NLP library. The first parameter refers to selecting one of the many available BERT models released by Google. We used the English language models and multilingual models only. These models were used as is, with no further fine-tuning for language modeling tasks as we want to evaluate the performance using the models as they are. The names of the models used are as follows: *bert-base-multilingual-uncased*, *bert-base-multilingual-cased*, *bert-base-uncased*, *bert-large-uncased*, *bert-base-cased*, *bert-large-cased*, *bert-large-uncased-whole-word-masking*, *bert-large-cased-whole-word-masking.*

The latter three parameters are used to determine if a question is answered correctly. For example, if the model is allowed 10 guesses and gets the answer correct on the sixth guess then the answer is considered correct. Similarly, if the answer is a synonym of the correct answer or is sufficiently similar to the correct answer, the answer is considered correct. For the purposes of this study "sufficiently similar" is defined as having a similarity score of more than 0.70. These parameters all have the potential to be used in the final language pipeline which produces the vocabulary set. The procedures of this study then aim to answer the following questions:

- Which BERT model works best?
- Will including synonyms or similarity scores improve the model's performance?
- Will looking at the first X guesses give better results? What number is X?
- What configuration of each of the above parameters works best for each the twelve student groups being considered?

The goal of this research is not to determine if the model can answer the questions correctly, but rather if the model can answer the questions in the same way as a human. The basic procedure is as follows: Have the model "answer" all questions in the data set, then compare the model's answers to the human's answers. If the model answers a question correctly, this question is easy according to the model, otherwise it is difficult. The definition of "correct" is different depending on the parameters being tested:

- An "exact" correct answer is when BERT guesses the exact word or a lemma of that word.
- A "similar" correct answer is when BERT guesses a word that has a 70% or higher similarity score according to the Spacey NLP library.
- A "synonym" correct answer is when BERT guesses a word that is a synonym according to the Merriam-Webster API
- An "any" correct answer is when BERT guesses a word that is correct according to any of the above measures

All of these correct answer types are tested repeatedly, allowing for each model to make 1-100 guesses. Because an extreme level of granularity is not necessary, the models are evaluated on guess-levels of size 10, so that the first 10 guesses are considered, then the first 20 guesses, etc. This results in 10 guess-levels.

After data is collected to determine the difficulty of words according to each configuration, those are compared to the difficult and easy questions for each human group. This is done once with the configuring set to determine which parameters are the best. Once the parameters are chosen for each student group, the parameters are tested with the test set to see if they still perform well.

## 6. Examining the Output of the VSGP

To examine what kind of vocabulary sets are generated for different student levels, we created vocabulary lists for three sample students and two sample texts, for a total of 10 sample outputs. The sample students belong in each level, advanced, intermediate, and beginner. We use the models configured for 60% difficulty at the respective levels since this was the threshold used by [8]. The first advanced student's vocabulary consisted of the first 5000 most frequent English words, the intermediate student at the first 3000 words, and the first beginner at the first 1000.

**Table 2: Description of Sample texts**

| | Word Count | Flesch Reading Ease | Flesch-Kincaid Level |
|---|---|---|---|
| Fiction | 3944 | 72.5 | 9.7 |
| News | 970 | 42.1 | 13.3 |

**Table 3: Model Performance vs Instructor Performance for All-Students set**

| Percentile | Model Sensitivity | Model Specificity | Instructor Sensitivity | Instructor Specificity | Specificity Difference | Sensitivity Difference |
|---|---|---|---|---|---|---|
| 60% | 0.83 | 0.59 | NA | NA | NA | NA |
| 50% | 0.64 | 0.66 | 0.57 | 0.89 | 0.12 | -0.26 |
| 25% | 0.86 | 0.26 | 0.85 | 0.69 | 0.01 | -0.62 |

To understand how changing the model type affects vocabulary, we have additional test students with which we controlled for vocabulary. These additional students consist of one a beginner and one advanced student, which have the same amount of vocabulary as the intermediate student. This is to compare students with different guessing abilities, grammar skills, and other non-vocabulary skills required for reading.

To get a sampling of different types of texts, we chose one fictional text and one news article. We have selected these texts to showcase different genres and lengths. They represent upper level, more difficult texts that we would expect students to need more support with. We show the word count, the Flesch Reading Ease Score and the Flesch-Kincaid Level, as calculated by Microsoft word. The Flesch Reading Ease Score goes from 0 to 100, where higher scores are better. Flesch-Kincaid Level shows the approximate grade level in terms of the US education system. See Table 2 for a summary of the texts.

We ran the Vocabulary Set Generation Pipeline once with each text and student. The model configurations for each student group matched those chosen from the model testing. The number of words in the produced vocabulary lists should give us an understanding of which students will have more words to study when using the VSGP; advanced students should need fewer words than beginning students. After generating the vocabulary lists, we will compare the number of words in the lists to the number of words in the text, specifically considering the number of words needed for minimal comprehension and comfortable reading. We hope to show that the number of words in the produced vocabulary sets is reasonable given the number of words in the text and the number of words originally known by the student. Since the source of synonyms used in testing limits API requests, the synonym functionality was tested for the model testing but not used by the pipeline.

## 7. Results

The results of the Model assessment and verification are presented in this section.

### 7.1 Criteria

In the Vocabulary Set Generation Pipeline, the model takes all unknown words in the text and attempts to determine which are difficult (true positive) and which are easy (true negative) so that difficult words can be added to the vocabulary study set. Since difficult words are considered unguessable, we know that too many false negatives may result in a reduction of student comprehension. Adding easy words to the study set (false positives) will not negatively affect comprehension, but it will reduce the number of words that students can guess in context. We would ideally like to maximize the number of words guessed in context because learning guessable words explicitly wastes time and removes a chance for practicing guessing skills. However, neither issue is as detrimental as missing a difficult word. Learning guessable words explicitly is not necessarily a waste of time as it will deepen students' knowledge of those words. Even though the words could have been guessed, the explicit learning still has value. If enough easy words are left out of the study set, students will also still have ample opportunities to guess in context, so it is arguable if absolutely all easy words must be removed from context or if removing a decent sized subset is good enough. On the other hand, the reason for glossing and pre-teaching is to make a text comprehensible to students, so failing at this goal is worse than including many extra words. For this reason, we chose to examine sensitivity, the ability for the model configuration to detect difficult words, and specificity, the ability for the model to detect easy words. Sensitivity is defined as the number of true positive results over the number of actually positive cases and specificity is the number of true negatives or the number of actually negative cases. Ideally, the sensitivity should be as high as possible, and the specificity should not be too low.

We use information provided by [12] to compare the model configurations to a human baseline. They surveyed three experienced test designers and university professors in their ability to predict the difficulty students will have with C-tests, a test that assesses similar skills as the cloze test. This data is a suitable measure of instructors' abilities to predict student difficulty in reading thus, we will compare the model configuration's ability to predict student difficulty in reading to this. Comparisons should be tempered by the fact that this shows only instructor ability to detect difficulty in reading *in general* and not with the same data set used to test the model configuration. The instructors were not asked to predict the difficulty of questions at different student levels, so we will only compare the sensitivity and specificity the instructors' performance to the model's performance with all students.

**Table 4: Results of the Model Configurations**

| Level | Percentile | Sensitivity | Specificity |
|---|---|---|---|
| Advanced | 60% | 0.79 | 0.75 |
| | 50% | 0.73 | 0.71 |
| | 25% | 0.65 | 0.61 |
| Intermediate | 60% | 0.67 | 0.45 |
| | 50% | 0.7 | 0.62 |
| | 25% | 0.69 | 0.46 |
| Beginner | 60% | 0.67 | 0.36 |
| | 50% | 0.88 | 0.3 |
| | 25% | 0.64 | 0.65 |

The instructors were asked to predict the error rates of a set of C-tests by classifying the questions as follows: Fewer than 25%, between 25% and 50%, between 50% and 75%, and more than 75% of students will get the question wrong. We will be comparing the sensitivity and specificity of the model configurations to that of the instructors.

## 7.2 Analysis of Results

Here, we examine the results of the best performing model configurations. Since we value sensitivity (the ability to find hard words) over specificity (the ability to find easy words) we attempt to choose model configurations which showed at least 0.75 sensitivity and 0.50 specificity in the configuring set, and prioritize improving sensitivity over specificity. However, should sensitivity reach at least 0.90, we prioritize specificity. Table 4 shows each level and the performance of the selected model configuration on the test set in terms of sensitivity and specificity. We also show the percent difference from the sensitivity and specificity of the instructors in the all-students group in Table 3.

The model configurations were able to reach the target sensitivity of 75% for four of the twelve student groups when tested with the test set. The target specificity of the 50% was reached for seven of the nine student groups when tested with the test set. Since 60% was the threshold used by [8] to determine if words should be glossed, we review the details of the best model configurations for each student level at the 60th percentile in Table 5.

**Table 5: Model Configurations chosen for each level**

| | Advanced | Intermediate | Beginner |
|---|---|---|---|
| Guess # | 60 | 10 | 50 |
| Answer Type | Exact and Synonyms | Exact | Exact and Synonyms |
| Model | bert-large-uncased | bert-large-cased-whole-word-masking | bert-base-multilingual-uncased |

## 7.3 Discussion

We believe that the results show potential for this method of student modeling via language modeling. The best results, in terms of balancing sensitivity and specificity, were in the advanced set. This indicates that BERT models may model advanced students better than other groups. In terms of identifying the most difficult words while also identifying a decent amount (50% or more) easy words, the best performing model configuration was that selected for all students with a difficulty threshold of 60%. Since the 60% threshold is the most difficult threshold, and since the all-students data set is skewed to the more advanced side, the all-students data set at the 60% threshold represents, an advanced group, which BERT seems to model well. Another success of the model configurations comes from the comparison to the difficulty detection ability of the human instructors. Keeping in mind that the results of the human instructors represent only general ability to detect reading difficulty, we see that the model performs about the same as instructors in terms of sensitivity. While these are only preliminary results, since direct comparison is not possible, we believe this indicates the possibility of human level or near human level ability to detect reading difficulty. While the specificity of the model is lower, the number is not so low that too many easy words would be removed from the text. The model also has the advantage of being more consistent because the human instructors were observed to not agree well on which words are difficult. The results are worse for the data set including only beginning students than the intermediate, advanced, or all-students sets. We believe this is because the questions in the beginner data set contained on average only about 2 responses. We also had only 170 questions for the beginner student group as opposed to 200 for the other groups. Thus, for the beginner set, we were unable to select the model that best fit the students' pattern of difficult and easy questions because the smaller data sets did not give any real pattern of how students perform.

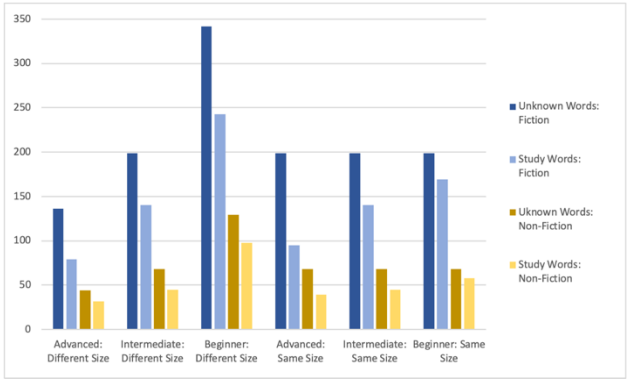


**Figure 2: Originally Unknown Words vs Words Chosen for**

## 7.4 Output of the VSGP

We tested the VSGP with three sample students of differing levels and vocabulary sizes, and three sample students of differing levels and same vocabulary sizes. We used the model configurations for the students at the 60%

level, except we did not include synonyms due to API restrictions. The model configurations with synonyms performed somewhat worse than with synonyms in terms of specificity, so we expect a to see slightly larger vocabulary lists than we would expect when using synonyms. The number of unknown words in each text and the number of words selected for study are shown in Figure 2. The percent of unknown words in each text and the percent of unknown words that are not part of the study set are shown in Figure 3. Finally, the overlap coefficient of the vocabulary sets for the students with the same vocabulary is presented in Table 6. The overlap coefficient shows what percentage of words in the smaller of two sets are shared in both sets. We would expect for students with the same vocabulary and different levels that the intermediate student's set is a subset of the beginner student's set (100% overlap) and that the advanced student's set is a subset of the intermediate student's set.

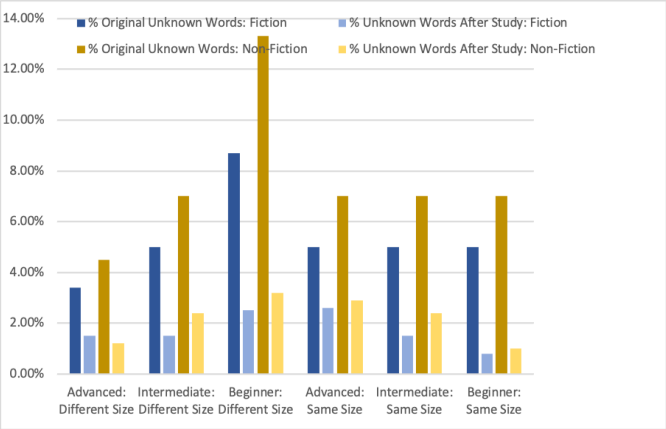


**Figure 3: Percent of Word Originally Unknown vs Percent of Word Unknown After Studying**

**Table 6: Overlap of Vocabulary sets**

| | Fiction Overlap | Non-Fiction Overlap |
|---|---|---|
| Beginner vs Intermediate | 0.91 | 0.98 |
| Intermediate vs Advanced | 0.93 | 0.92 |

## 7.5. Discussion of VSGP output

These results show vocabulary sets generated for students of differing levels. When controlling for vocabulary knowledge, we see that more advanced students receive fewer study words than less advanced students. The overlap coefficient is above 0.90 for all students of the same sized vocabulary and both texts, showing that, for the most part, the study sets of more advanced students are subsets of the sets for less advanced students.

At first glance, it may appear that the number of words to study is very large, however, we must account for the student's vocabulary knowledge and language level, as well as the length of the text. Analysis of the percent of remaining unknown words indicates that the vocabulary lists produced for most students are about the correct size. The lists can give students enough context to comprehend the text and enable them to guess the remaining words. Recall that between 95% and 98% of words are required for reading comprehension like that of a native speaker, with 98% indicating comfortable reading levels. Observing the percent of words that are unknown in the text versus the percent of words in the text left for implicit learning (the unknown words not chosen for study) we see great results. For the selected texts, the percent of unknown words was usually too high for good comprehension. In some cases, it was at or around the 5% threshold of unknown words which allows for comprehension but makes reading difficult. For all students, the percent of words left to guess after studying the given vocabulary set was closer to 2% than 5%, and for most it was near 2%. For only one student, the beginner with mid-range vocabulary, remaining vocabulary might be worryingly low. However, the results from the model configuration analysis indicated that the beginner level at the 60% threshold includes too many easy words, which is likely the cause of the low remaining vocabulary. These study sets can be used for vocabulary pre-teaching or text glossing to support students in implicit language learning because they enable comprehension but do not explicitly teach so many words that the student has no words to guess in context.

## 7.6. Limitations and Future Work

The selection and testing of model configurations was limited by the small amount of data available for students at lower levels. Conducting a new study to gather more responses would produce data that creates a better pattern of difficult and easy questions to which the model can be better configured. A set with more questions would also help to better configure and compare the models more accurately. Better data on instructor ability to identify difficult words would also improve comparisons to the expert baseline. Additionally, a user study should be performed to verify the model with real users. Additionally, other models such as bi-directional LSTM ULMFiT model [16] or the modified transformer model [21] which uses an LSTM with a transformer could possibly yield good results in modeling student reading comprehension. It may also be possible to use smaller pre-trained models and fine tune them with learning materials. Another possible improvement might be to train models from scratch using only data that language students would be familiar with.

# 8. Conclusions

We discovered model configurations for the masked language modeling models provided by BERT that have the potential to model students' reading abilities. The good

results on the all-students data set at the 60% threshold and advanced-students data set shows that this method is useful for modeling student comprehension at more advanced levels. This threshold was chosen by expert opinion as a good threshold for choosing words that cause difficulty for students in reading. With the model configurations, the VSGP produces vocabulary sets of reasonable size, giving students enough explicit vocabulary instruction to read comfortably while also leaving enough words left for implicit vocabulary acquisition. This method also has the potential to be applied to any text. The VSGP achieves this without requiring much data about the student's language knowledge besides general language level and known vocabulary. It is open source and designed to be flexible, allowing it to be easily integrated into new language learning systems or used to build new software. This is vastly different from existing applications which require the user to begin learning before any adaption occurs and use results recorded by millions of students to determine difficulty level. Instead, the VSGP is designed to provide good results from the very first time a user uses it, whether the user is the application's first or millionth.

## 9. References


[1] B. Laufer, G. C. Kalovski, "Lexical Threshold Revisited: Lexical Text Coverage, Learners' Vocabulary Size and Reading Comprehension," *Reading in a foreign language,* vol. 22, no. 1, pp. 15-30, 2010.

[2] C. R. Heil, et al, "A review of mobile language learning applications: Trends, challenges, and opportunities," *The EuroCALL Review,* vol. 24, no. 2, pp. 32-50, 2016.

[3] I. S. P. Nation, Learning vocabulary in another language, Cambridge: Cambridge University Press, 2013.

[4] M. S. Swanborn, K. Glopper, "Incidental word learning while reading: A meta-analysis," *Review of educational research,* vol. 69, no. 3, pp. 261-285, 199.

[5] J. P. Loucky, F. Tuzi, "Comparing foreign language learners' use of online glossing programs," *International Journal of Virtual and Personal Learning Environments (IJVPLE),* vol. 1, no. 4, pp. 31-51, 2010.

[6] H. Healy, "Dictionary use," *The TESOL Encyclopedia of English Language Teaching,* pp. 1-7, 2018.

[7] B. Zimmer, "Science of Learning," 2015. [Online]. Available: https://www.vocabulary.com/educator-edition/Vocabulary.com-Science of Learning.pdf.

[8] M. H. Ko, "Glossing and second language vocabulary learning," *Tesol Quarterly,* vol. 46, no. 1, pp. 56-79, 2012.

[9] Z. Ÿ. Ertürk, "The Effect of Glossing on EFL Learners Incidental Vocabulary Learning in Reading," *Social and Behavioral Sciences,* vol. 232, pp. 373-381, 2016.

[10] J. Hill, R. Simha, "Automatic generation of context-based fill-in-the-blank exercises using co-occurrence likelihoods and Google n-grams," *11th Wkshp on Innovative Use of NLP for Building Educational Apps*, San Diego, 2016.

[11] M. Felice, P. Buttery, "Entropy as a Proxy for Gap Complexity in Open Cloze Tests," in *International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, Varna, 2019.

[12] L. M. Beinborn, Predicting and manipulating the difficulty of text-completion exercises for language learning, Technische Universität Darmstadt, 2016.

[13] L. Beinborn, et al, "Predicting the difficulty of language proficiency tests," *Transactions of the Association for Computational Linguistics,* vol. 2, pp. 517-530, 2014.

[14] W. S. De Mulder, S. Bethard, M. F. Moens, "A survey on the application of recurrent neural networks to statistical language modeling," *Computer Speech & Language,* vol. 30, no. 1, pp. 61-98, 2015.

[15] J. Devlin, M. et al, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805,* 2018.

[16] J. Howard, S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146,* 2018.

[17] G. Zweig, C. J. Burges, "A challenge set for advancing language modeling," in *NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, Montréal, 2012.

[18] L. Beinborn, et al, "Difficulty Prediction for Language Tests," Technical University of Darmstadt, 2014.

[19] L. T. Beinborn, et al, "Candidate evaluation strategies for improved difficulty prediction of language tests," in *the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, Denver, 2015.

[20] H. Liu, M. Cocea, "Semi-random partitioning of data into training and test sets in granular computing context," *Granular Computing,* vol. 2, no. 4, pp. 357-386, 2017.

[21] C. Wang, et al, "Language models with transformers," *arXiv preprint arXiv:1904.09408,* 2019.

[22] ExplosionAI: spaCy Industrial-strength Natural Language Processing in Python. Available: https://spacy.io

[23] Google-Research, "google-research/bert," Available: https://github.com/google-research.