Evaluating Utility of Vocabulary to Language Learners

Leonard Paál
Hochschule Karlsruhe
Supervising professor: Jannik Strötgen
Additional supervision by: John Blake (University of Aizu)

December 2024

Abstract

[Abstract]

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

1.1.1 Role of Vocabulary in Language Acquisition

Learning a second language involves many different skills, often categorized into listening, reading, speaking, and writing. Another categorization may be vocabulary, grammatical skills, the ability to understand known words in various accents, understanding language when spoken at a fast speed. One skill that is required for any of these if the knowledge of vocabulary in the target language. A person with basic grammatical skills but no vocabulary has no ability to express themselves or understand anything which they hear around them. On the other hand, a person familiar with rudimentary vocabulary but no grammatical knowledge may struggle with understanding complex sentences and sound unnatural when speaking, but can at least make sense of short phrases and express themselves. Thus, basic knowledge of vocabulary is clearly one of the most essential skills for using a language. This raises the question of which vocabulary should be learned first when starting out on the journey of language acquisition. This work attempts to contribute to answering this question.

ToDo: mention various methods of finding useful vocabulary from viewpoint of learner: textbooks, apps, etc.

1.1.2 Context-specific Language Learning

Learners of languages are typically interested in one or more specific aspects of the language. There is no such thing as an unbiased corpus than can fit every use case. Decisions must be made how much focus is given to everyday conversation, academic writing, writing pertaining to business like job applications etc. Many textbooks group vocabulary by topic, with a new topic being introduced with each lesson that student should ideally be able to converse in after completing the lesson. However, this way of introducing vocabulary has several shortcomings:

- Students spend time learning specific terms about the topic in one lesson while not learning even general vocabulary in other topics until much later.
- Knowing the words from previous lessons becomes a prerequisite for the more advanced material, especially because the terms from earlier lessons are used in example sentences for grammar. Thus, learners interested in learning the use of the language in one context will have a hard time of skipping earlier, less pertinent lessons to them.

Since the advent of computer-aided natural language processing, methods have been suggested to computationally identify useful words: Nation and Waring propose in an 1997 study [16] that the frequency with which a word appears in the target language should be used a metric for its importance to the learner, or utility, and thus teaching high-frequency words should be the focus when teaching beginners. However, focusing on maximum-frequency words to achieve text coverage (e.g., knowledge of 90% of words in a text) may not be as useful as one might at first think, as the most frequent words tend to be generic terms like “but”, “from”, “time”, “world” etc. Knowing only these words is not sufficient for comprehending most texts.

The TF-IDF metric [18] is often employed for finding the keywords in a document and thus a proxy for how important a word is for the overall meaning of the document. It essentially is a frequency normalized against a larger, background corpus and expressed whether the usage frequency in the document is unusual. Using TF-IDF as a measure for utility of a word in a given context is possible, but it suppresses words that may be generally useful.

Thus, there is a need for further exploration as to how word utility can be calculated, using modern NLP methods involving artificial intelligence where necessary. Put more simply, this question may be reduced to:

What order or words, when learned, gives the learner the best set of words to understand and communicate in the language as quickly as possible?

1.1.3 Examples of Contexts and Words

Some examples for contexts that could be interesting to language learners include:

- Reading Wikipedia articles about a specific field (computer science, literature, biographies)
- Watching movies
- Travel to a country where the target language is spoken
- Doing business with a company from a country where the target language is spoken
- Cultural exploration (literature, religion)
- Finding friends from other countries

The different contexts for which learners might be motivated to learn a language differ in how easily corpora can be obtained about to mine patterns from. Movie subtitles and Wikipedia articles are easily obtained from sites such as opensubtitles.org and wikipedia.org. The words that might be relevant for travel are not as easily obtained: One might imagine an ideal scenario to collect data, in which a statistically relevant group of people travelling to the destination to be examined are randomly selected and equipped with microphones and cameras before the travel. During travel, one could record their conversations, conversations with people around them, and materials they attempt to read to navigate their journey such as train schedules, descriptions of tours, restaurant menus, street signs, etc. Lacking the funds to conduct an experiment for every possible language, this paper is interested in finding a methodology to obtain data from readily available corpora and websites online that extracts relevant vocabulary from the source texts. Depending on the context, we can think about which of the following English words might be likely to appear frequently in the texts:

- Convert
- Cash

- Hug
- Dammit
- Y'all
- From
- Nineteen eighty four
- Married

To examine a few examples: Words like "convert" occur frequently when looking at Wikipedia articles [1]. "cash" is likely to be useful for travelers, but in most other contexts, it would not be as relevant. "Y'all" is almost never used in formal writing but used abundantly in everyday speech in the southern United Stated of America and South Africa. "From", meanwhile, will be likely to be one of the most frequently used words regardless of context.

[1] see "Wikipedia" corpus 2016, drawn from one million lines on https://wortschatz.uni-leipzig.de/en/download/English

1.2 Challenges and Contributions

1.2.1 Challenges/Research Questions

[What is the problem to be solved? - Given a context, find vocabulary maximally useful for understanding texts in it] Criteria for good extraction method:

- Requires little manual effort to generate input data (ideally performable on freely available corpora)
- Requires little computational effort
- Outputs word utilities that align with human intuition

Inherent challenges of text-based language anlysis include:

- ambiguity
- Words having multiple different meanings (polysemy). E.g. *can* can be a verb or a vessel.

1.2.2 Contributions

[Motivation: Useful vocabulary -¿ Need approach to evaluate utility of word] [How this paper address the challenges: Simulate human trying to understand texts with AI] [Distinguish between passive and active utility]

1.3 Outline of Work

Chapter content: explain the logical flow of the thesis chapter to chapter

Chapter 2

Background

This chapter gives context to estimating the utility of vocabulary.

We first go over how this work connects to research about vocabulary acquisition for second language learners in chapter 2.1.

After this, we defining the exact aim of the work in chapter 2.1.2.

The following three chapters explain the (partially overlapping) fields of Natural Language Processing, Machine Learning, and Explainable AI, tools from all of which will be employed in our word utility evaluation approach laid out in chapter 3.

Finally, we examine current approaches for selecting vocabulary for language learners in chapter 2.5, to show what methods are currently employed and how they could be improved.

2.1 Context of the Work

Chapter content: language learning in general, cite reference works and textbooks. Maybe mention context-specific learning resources too, such as academic English

2.1.1 Linguistic motivation

The issue of which words to learn when setting out to acquire a language is not a new one. Every textbook on language which does not focus exclusively on grammar must ask address the issue, and proactive learners will doubtless find themselves finding ways to optimize the return on their invested studying time.

The Routledge Handbook of Second Language Acquisition [14] gives three criteria for deciding the order in which words are taught to beginners:

1. Frequency
2. Usefulness
3. Easiness

The first criterion, frequency, is easy to justify: Learning words that appear with great frequency will allow the learner to understand the maximum percentage of wrods in texts, and should thus be among the most relevant. The *easiness*, or *learnability* of a word is defined as how easy it is for a learner to acquire the word. The authors define it with respect to an individual learner since it is posited that the various linguistic backgrounds of learners will influence which words they can acquire with ease. Apart from learner-independent factors influencing the learnability of a word such as word length, is sure to have an impact whether the learners already knows a cognate of the word:

We can imagine a native German speaker attempting to learn English: The word "internationalization" may not be a particularly frequent word in most contexts and thus not seem too important to teach, but the German will likely recognize the word as a cognate of the German equivalent "Internationalisierung", and acquire the word with ease, giving a justification for teaching the word early if we wish to impart the most language knowledge as quickly as possible.

Finally, there is the criterion of "*usefulness*". While it is distinguished from frequency, the authors do not define what constitutes usefulness, or how it might be measured.

In their 2019 paper [7], He and Godfroid pick up the above three criteria in order to group words into clusters which should help determine their priority in vocabulary learning. However, again it is not defined what constitutes usefulness and no way to measure it is suggested beyond "human intuition". In fact, it is put in contrast with frequency, which may be derived from corpus data, whereas usefulness cannot.

As the aim of this work is to find ways to evaluate word utility automatically, the following chapter will attempt to define utility and related concepts more carefully, in order to state the aim of this work more formally.

2.1.2 Statement of Goal

Chapter content: Statement of goal, definition of "utility", context

Generally speaking, this work addresses the problem of evaluating how useful it is to learn a given word in the target language of a language learner. This helps us in generating ordered lists of vocabulary to learn which aid the learner in gaining competency in their target language quickly. Since different language learners learn languages for different reasons [Missing citation about: Statistics on motivations of language learners], we attempt to take into account the learner's motivation to tailor vocabulary specifically to the individual.

To make this goal more specific, we define several concepts that help us speak more precisely about this goal and how it may be accomplished. These concepts are used to throughout this work.

Linguistic Context

By a linguistic context or just *context*, a subset of situations is meant in which a person interacts with language. Examples of language contexts by topic are: News, football, crocheting.

Examples of language contexts by situation are: Everyday conversation, scientific articles, watching movies.

Contexts can be further subdivided into smaller contexts, or grouped across different lines: "News" could be subdivided into "international politics", "sports", "business", etc.

The concept of a language context is useful to a language learner since it enables them to prioritize learning vocabulary and grammar associated with the context they are interested in: Someone who is learning Arabic to better understand middle-eastern politics will have little use for words which are mostly associated with football.

The context could even determine the complexity of sentences that a learner must be able to understand: Some contexts, like everyday conversation, can be navigated fairly well with short sentences, but political speeches often feature long and structurally complex sentences, with various relative clauses, multiple negations, etc. [Missing citation about: Complexity differing across contexts] Thus, learning for one or multiple specific contexts gives the learner a more concrete goal and thus better idea of what they must learn to achieve it.

In contemporary methods and tools for language learning, usually little emphasis is placed on learning for a specific context, or else few contexts are available [Missing citation about: investigage current popular language tools. Classroom cannot cater to individual preferences and apps only allow some personalization].

Utility

In this work, *utility* is defined as the increase of language ability in a given context that a person gains by knowing a word versus not knowing it at all. Here, language ability refers to being able to understand more of texts they read in the given context, being able to speak about the subject, etc. It is easy to see that some words will be more useful than others given a context: In the context of international news,

learning the word "war" will bring more understanding to the learner than "pear" or "polymorphism".

Proxy Task

Some of the concepts in the realm of language learning cannot be measured directly: This can be because they are psychological in nature ("understanding") or because they are too complex to be objectively measured by numbers ("language ability"). The above section defines word utility as an increase in "language ability", but to analyze this ability with computers, we must make be able to put a number on it, even though we lack both an agreed upon scale and unit of measurement.

To circumvent this issue, so-called *proxy tasks* will be used: Proxy tasks are tasks that test subjects can perform, where the test result (the *proxy measure*) is used to capture the abstract quantity under study [Missing citation about: Proxy task definition]. For example, as a proxy measure for a person's general spelling ability, we could make them choose from multiple spelling variants of a word, and use the accuracy with which they select the correct answers.

This work employs AI models solving NLP tasks as an economic and repeatable substitute for real humans interacting with language, and uses the NLP tasks as proxy tasks to make "language understanding", and thus "utility", measurable concepts that can be computationally maximized.

2.1.3 Summary

With the concepts above defined more precisely, let us state the goal more accurately:

To find words that have the maximum *utility* given a particular *language context* by means of *proxy tasks*.

With the words in cursive filled in, this translates to: Finding words that provide a language learner with the most language understanding in the specific situations they are interested in, requiring them to learn the least possible amount of words. Since the understanding of a learner cannot be directly measured, we use tasks that check the learner's understanding as an imperfect but necessary approximation.

The following chapters will go over the fields addressed in achieving this aim, and highlight relevant aspects from them that contribute in later chapters to conceptualizing and implementing a solution.

2.2 Natural Language Processing

While the aim of the work is to make vocabulary lists that serve human learners, the technical implementation of this necessarily will process human language with the use of computers. This domain is called *Natural Language Processing*, a field defined in The Handbook of Computational Linguistics and Natural Language Processing as the engineering domain of Computational Linguistics [2]. Natural Language Processing, often abbreviated as *NLP*, is both used in Computational Linguistics (the field concerned with analyzing natural language through language data) and as well as in practical applications such as Chatbots, Machine Translation, Speech Recognition etc. [9]. This work addresses of these both aspects, as it is an undertaking of Computational Linguistics through the analysis of how AI models perform typical **NLP tasks**.

2.2.1 Corpus

The typical way Computational Linguistics empirically analyzes language is through the use of **corpora**. A common definition of a corpus can be seen in [8] as "an electronically stored collection of samples of naturally occurring language", which "can be a test bed for hypotheses and can be used to add a quantitative dimension to many linguistic studies".

Recently, many large corpora have been compiled and are freely available to the public. Corpora differ from each other mostly in their source and method of compilation. Depending on their source, some corpora may serve as examples of language being used in a language context, such as news or movie subtitles.

In the implementation and evaluation of this work's approach to word utility estimation, several such context-specific corpora will be used, as they present a way to analyze how language (and more specifically, vocabulary) is used in the linguistic context of the corpus. ToDo: Quote previous work that uses corpora to model linguistic contexts for the purposes of language learning

2.2.2 NLP Task

This work attempts to make use of not only static data in the form of corpora, but also the interaction of AI models with those corpora in order to gain insights into word utility. This active part of Natural Language Processing consists of performing **NLP tasks**. Typical NLP tasks include [9]

- Sentiment Detection: Given a text, estimate the emotional state of the author.
- Masked Language modeling: Given a text with a word blanked out, estimate what the word should be.
- Machine Translation: Given a text, translate the meaning into another language while preserving meaning as faithfully as possible.

Humans are not trained from childhood to do any one language "task": They can have conversations, answer questions about things they have seen, etc. In contrast, the interaction of computers is usually more rigidly defined by specific NLP tasks, and AI models are trained to perform one or several of these tasks.

For the purposes of this work, an *NLP task* is a function with a specific input and output format, where at least one of the two formats takes the form of natural language. NLP tasks will serve as a way for AI models to interact with natural language, enabling us to analyze how certain words influence this interaction; the exact approach is described in chapter 3.

2.2.3 NLP Preprocessing: Tokenization

Language presents itself in continuous form in most situations: When listening to spoken language, it is not obvious where one word ends and another begins. Likewise, written texts we find online or in books are not necessarily subdivided into its semantic constituents. While words in the written English language are mostly separated by spaces, a writer may choose to create a new hyphenated word sequence on the spot as necessity demands.

Further complicating the issue of where to separate words is the fact that many non-European language do not use spaces in their spelling (e.g. Japanese, Mandarin Chinese) or use spaces for a different purpose (separating syllables in Vietnamese, separating sentences in Thai). For this reason, *tokenizers* are used in Natural Language Processing to divide continuous texts into their words [9]. ToDo: Explain various types of tokenization?

Splitting continuous text into distinct words has several benefits:

- we can make statistics from them (e.g., counting which words occur many times).
- we can assign values to them, such as estimated utility.
- we can mask them in text inputs to AI models to test what effect masking a particular word has on the output.

In recent years, major advances have been made in the field of Natural Language Processing through the use of AI models. As this work makes extensive use of AI models, the next section therefore briefly introduces the field of Artificial Intelligence.

2.3 Artificial Intelligence

Elaine Rich, in her 1983 work "Artificial Intelligence", defines Artificial Intelligence as "the study of how to make computers do things at which, at the moment, people are better" [21]. An introductory article by IBM describes it as "a technology that enables computers and machines to simulate human learning, comprehension, problem solving, decision making, creativity and autonomy" [1]. And this This is opposed to traditional algorithms which can only follow algorithms which have been explicitly programmed.

2.3.1 AI Models

The building block of Artificial Intelligence is called an **AI Model**, which is a stochastic model using training data to learn patterns, and then used on inputs that were not identically present in the training dataset.

AI models are used in this work to simulate an agent interacting with language. The (pre-trained) AI model is thought of as a test subject in possession of linguistic skills that can be studied. Through its training process, AI models learn patterns that are generalizable across the problem domain. If these patterns resemble the knowledge that humans gain when they learn a new ability, it may be possible to analyze the interaction of AI models with data so that a human learner can learn from their behavior, similar to how humans can observe experts in their domain and learn from their behavior. The field which provides theses methods of analysis is Explainable AI, which is explained in chapter 2.4.

The current de facto standard for recent AI models is the *Neural Network* architecture [9], which attempts to imitate the neural makeup of the human brain. Such models achieve state-of-the-art results on most NLP tasks [Missing citation about: neural networks performing NLP task] and all models used in this work are Neural Networks for this reason. However, within this architecture there exist further subtypes of AI models, the most important for this work is the **transformer model**.

2.3.2 Transformer Model

Since their invention in 2017 [24], *transformer* models have brought impressive gains in performance to many fields, including Natural Language Processing.

They are a particular type of deep neural network characterized by an attention layer before the fully connected neural layers. Said attention layer makes the model "focus" on important parts of the important, while "ignoring" less important ones. This helps the model find patterns in noisy input. This attention is one possible way to attribute importance to inputs, and thus utility to words in our case. This use will be discussed in chapter 4.3.

This section has introduced the important AI models used in this thesis, namely the Neural Network and one of its most effective subtypes, the transformer. While these types of AI model have driven many recent improvements in performing common NLP tasks, deep neural networks stop being readily understandable to humans rather quickly once the number of neurons and layers is increased. We are interested in

[1] `https://www.ibm.com/think/topics/artificial-intelligence`, last accessed February 26th, 2025

how the inputs (words) influence the outputs of AI models when performing NLP tasks. This kind of analysis is the domain of **Explainable AI**, which is why the next section briefly introduces this last necessary field for the approach put forward in this work.

2.4 Explainable AI

Explainable AI is the field of research focused on making the decision-making progress of AI models more transparent and understandable to humans, to enable us to reason about the AI model's decisions [25]. This can be useful to check if the decision process contains social biases, or if it is based on wrong patterns learned from skewed training and test data (overfitting). By analyzing the decision process of high-performing AI models, this work attempts to extract useful information about how the language is processed by them which may be useful to humans as well. For this analysis, XAI is used a tool to find out which inputs influence the performance of AI models the most.

2.4.1 Distinctions between XAI methods

There are several categories of XAI methods that will be relevant in this work, because they determine not only the purpose of the method, but also which XAI method can be used on which AI model. This chapter will go into some of distinctions and argue why some types are more appropriate to the aim of this thesis than others.

As was mentioned before, this work is interested in analyzing the influence of words in AI model input on its output for the purpose of finding out which words might be the most useful for human learners as well. The question of which inputs are the most important for a model to reach its output is the domain of **feature importance explanations**, a sub-field of XAI concerned with the question: *"how does each feature affect the model?"* [2]. Thus, all XAI methods in this paper belong to this group.

Feature importance explanations can be further subdivided into two groups: **Feature selection** methods explain the model's decisions by trying to predict a subset of all input features which are deemed "important". **Feature attribution** methods attribute a concrete importance score $a_i \in \mathbb{R}$ to each input feature x_i. This work uses feature attribution methods exclusively, as the scores will allow us to rank words in a list by their importance.

Another important characteristics of an XAI method is whether it is **model-agnostic** or **model-specific**. A model-agnostic XAI method can be used no matter what AI model is used, whereas a model-specific method is limited to being used to a specific type of AI model, such as transformers, decision trees [Missing citation about: decision tree], or neural networks. Model-agnostic models are also called black-box-approaches, as they do not look into the model (such as weights in neural networks) but only perturb the inputs and observe changes in the model output. A advantage of model agnostic explanations is that they can be used on any AI model However, many recent state-of-the-art AI models are build using the transformer architecture, hence methods specific to transformer models can also be fairly broadly applied. A major upshot of model-specific explanations is their efficiency: Black-box approaches require us to perturb the input to observe differences in the output, requiring us to run the model multiple times for a single explanation, which can constitute a great computational effort. On the other hand, model-specific methods typically only require one run of the model, as they can look into the model itself for an

[2] `https://courses.cs.washington.edu/courses/csep590b/22sp/files/lectures/lecture2_part1.pdf`, last accessed on February 27, 2025.

explanation. For these reasons, both model-agnostic and model-specific approaches will be used in this work.

Another feature of XAI methods is the scope of its explanations: **Local** XAI methods explain why a model output was generated for one particular input. A **global** method attempts to reason about the model's behavior in general, independent of any particular input data. This work is interested in observing the interaction of the AI model with a particular corpus, not only the AI model. For this reason, all methods used in this paper will be local methods. However, while local explanations only explain a single decision of the model (which, in the work, will be the interaction of the model with a single sentence), we will aggregate the individual decisions across the corpus to get a bigger picture about this interaction.

The transformer attention mechanism introduced in chapter 2.3.2 has been used as a model-specific XAI method [13] [12] , and will be used in this work as one among several XAI methods employed to gauge the impact of words in the input to the output of (transformer) AI models. The use of attention as explanation, too, will be discussed in more detail when describing the implementation of our approach in chapter 4.3.

Table 2.1 summarizes the used of XAI in this work.

Attribute	**Type Selected for this work**	**Reason for selection**
Model-Specificity (Model-Agnostic or Model-Specific)	Both Model-Agnostic and Model-Specific	Model-Agnostic methods: Broad applicability; Model-specific methods: Computational efficiency
Importance Explanation (Feature attribution or Feature selection)	Feature Attribution	Use of scores to order words in vocabulary lists
Scope (Local or Global)	Local	Attain explanations taking into account the corpus

Table 2.1: XAI methods used in this work

2.5 State of the Art (of Vocabulary Selection)

2.5.1 Non-computational Methods

How is vocabulary order selected by current language teaching tools: Not context-specific in most cases offers only one order, investigate if there are any automatic approaches besides frequency]

[can present methods for extracting useful words from text] can be similar in either method or goal. goal is finding useful words for language learning method is extracting important words from text via XAI

[7] cites concepts of frequency, usefulness and difficulty of words as widely accepted criteria to decide when to teach it. However, the literature is not clear on how usefulness is defined outside of "human intuition" [7]. Furthermore, usefulness is presented as independent from frequency, making it more unclear still how it may be defined. The advantage of human intuition is that humans can understand the nuances of texts better than computers, especially before the invention of large AI models tackling NLP tasks. Relying on human intuition to evaluate word utility has two main drawbacks compared to computational methods:

- Difficult to put a concrete number on word.
- Evaluating many languages would necessitate human experts for each language, necessitating expensive studies.

2.5.2 Computational Methods

These methods generate vocabulary lists by using corpora and computer-aided language processing to compile vocabulary lists. They are closer related to the methods that this work strives for, and will be used as points of comparison when evaluating our own approaches.

Raw frequency of words in corpus A simple ordering of words by how often they appear in a corpus.

Frequency with stopwords filtered out The same as frequency, but filtering out known stopwords from the resulting lists.

TF-IDF How often the words appear in a target document but divided by their frequency in a more generic corpus. This metric is typically used to employ the most relevant words in documents for identifying keywords that express best its core topic.

2.5.3 Issues with Current Methods

Current methods do not exploit recent developments in AI technology and thus suffer from several shortcomings: In general, all of them essentially only count words without taking into account their relationship between each other:

Frequency: The most frequent words in texts are often words that carry little meaning by themselves, such as "a", "the", "of" in English. While these may appear in many texts, they are not useful in determining their meaning. TF-IDF: This

metric has been used successfully to find the words that give the best hints at a text's topic. However, it does not take into account and semantic relationships between the words in a text. Thus, learning words by aggregating TF-IDFs on multiple texts may aid in identifying the topic of texts, but not at finding out what the message conveyed about the topic is. "not" is a highly frequent word in English and thus will have a low TF-IDF score in most documents. But it is essential to know, as it can completely invert the meaning of a sentence.

Chapter 3

Approach

This chapter describes in detail our approach for word utility evaluation to fulfill the goal stated in chapter 2.1.2.

The problem is first put formally by putting the terms of word utility and vocabulary list efficiency into mathematical objects in chapter 3.1. We then describe a hypothetical experiment that could be performed to evaluate word utility using human feedback in chapter 3.2 After pointing out the unfeasibility of such a testing method, we suggest ways to perform the experiment with AI models instead of humans as the test subject in chapter 3.3, using the technological building blocks referred to previously in chapter 2.

With a method for utility evaluation established, we then point out why it is in itself not yet sufficient for actually generating useful lists of vocabulary that a human learner could utilize in chapter 3.4. To solve this final problem, 3.5 puts forward a general approach for list generation.

3.1 Formal Problem Statement

To repeat the essential sentence of chapter 2.1.2: The aim of this work is to find words that have the maximum *utility* given a particular *language context* by means of *proxy tasks*.

Keeping in mind that this is done in order to sort words into vocabulary lists that can be used by language learners, we can conclude that the output of a proposed solution to this problem would be an **ordered list of words**. Theoretically, if this word list is ordered by descending word, we can speak of a **maximally efficient** vocabulary list. We could also put a number on *any* list of words which measures how efficient a vocabulary list is to understand a given linguistic context. A list should be evaluated as efficient if useful words appear at the top of the list, and less efficient if less useful words are at the top, since that would mean that someone learning the words in that order would not gain understanding at the highest possible rate.

We can subdivide the overarching goal of word utility evaluation into three related, but not quite equivalent sub-goals:

- Evaluating the utility of a single word.
- Evaluating the efficiency of a list of words to learn.
- Generating maximally efficient lists of words to learn.

This section will put utility and efficiency into mathematical terms, to provide a clearer explanation of the variables involved in these tasks and their relationship.

Proxy Task

Since utility is defined in terms of language ability, we first need a way to put a number on the language ability of a test subject. This is done by means of a proxy task: We can imagine a human test subject taking a language exam as a proxy task to find out their language ability:

$$\begin{aligned} t : \text{Human} &\to [0, 1] \\ t(s) &\mapsto p \end{aligned}$$

where s is the test subject and t is the proxy task, with a possible performance score p between zero and one.

We can imagine many variables going into this function, such as the time when the test is taken (hopefully the human's performance would increase over time). However, this thesis addresses vocabulary learning, and so the vocabulary of the test subject is provided as an additional parameter to the function t to make a slightly modified function, t':

Proxy Task with Vocabulary

$$\begin{aligned} t' : \text{Human}, 2^W &\to [0, 1] \\ t'(s, V) &\mapsto p \end{aligned}$$

where W be the set of all words in the language, and $V \in W$ is the vocabulary of the test subject.

Vocabulary List Efficiency

We now have a function t' from the vocabulary to the score in the proxy task (the proxy metric for language ability). Using this function, we can define a measure for the efficiency of an ordered list of vocabulary l containing elements from W.

A list of vocabulary is simply an ordered list whose elements are words of the target language, with no duplicates:

$$l := (l_1, l_2, \ldots, l_n) \quad \text{where } l_i \in W \text{ and } \forall i, j : i \neq j \implies l_i \neq l_j.$$

Note that by this definition, the list need not contain all words of the language.

If the test subject learns the word exclusively in order of the list l (and retains them perfectly), their vocabulary at any point will be that list up until some index k:

$$V_{l,k} := \{l_i \mid i \leq k\}$$

This thesis aims at finding vocabulary lists that improve the ability of a language learner at the fastest possible rate. We can call this property the efficiency of the list. To measure it, we can think of a score $p_{aim} \in [0, 1]$ that the learner wishes to achieve in the language exam. An efficient list will let the learner achieve their goal in the shortest amount of time, or the least amount of words (see figure 3.1). Therefore, the efficiency of a vocabulary list shall be inversely proportional to the number of words which must be learned from the vocabulary list to bring the performance above the threshold p_{aim}:

$$e_{t,s,p_{aim}}(l) \mapsto -\min\{k \mid t'(s, V_{l,k}) \geq p_{aim}\} \tag{3.1}$$

And with this definition of efficiency, we can define the condition for an optimal vocabulary list given a particular p_{aim}:

$$l_{opt,t}(s, p_{aim}) = \arg\max_{l} e(s, l, p_{aim}) \tag{3.2}$$

Finding vocabulary lists that approximate an optimal list according to formula 3.2 is the aim of the rest of this work.

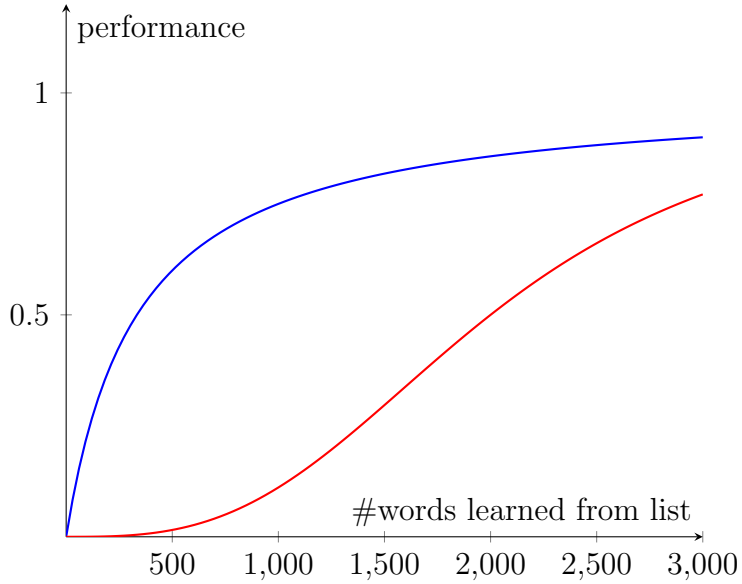

Figure 3.1: Visualization of vocabulary list efficiency: **Blue** represents the learning curve with an efficient list, **Red** represents a less efficient list (i.e, less useful words at the top of the list)

3.2 Experimental Setup: Measuring Word Utility as Ability Improvement in Humans

Chapter content: state utility in terms of language ability improvement aim is not just to evaluate, but also efficiently find lists of vocab give problems that is faced when trying to test this approach with human test subjects: no repeatability, high costs

With the formula 3.2, we now have set our goal more concretely. We are left with the question of how to design an actual experiment to test vocabulary list efficiency, and also how to generate one that is approximates an optimal list. Hence, this chapter lays forth a hypothetical experiment design by using a human test subject to test efficiency, which will however be found to suffer from problems of feasibility, which the following chapters will set out to solve.

Experimental Setup

Let us first consider an example scenario, where we wish to measure the efficiency of one vocabulary list l_A of 3000 words for a desired performance of $p_{aim} = 0.8$:

We would need to find a human test subject s who does not know any words of the target language at the outset. We would then make the subject learn the words from l_A one by one, and test their language ability in regular intervals to see when their performance reaches 0.8. This disregards the fact that, depending on the task, even knowing all words in the language would not necessarily mean the performance would reach 0.8, but assuming a normal school test that is not an issue.

Problems of the Setup

While this test setup might take a long time to complete (especially for higher thresholds), it is still feasible. But if we wish to compare the efficiency of l_A with a second list l_B, we must measure how the same subject performs if they learn from the second vocabulary list, which presents is an unavoidable issue: **The test subject cannot arbitrarily forget the words they have learned from the first list l_A.** The experiment is thus not repeatable.

If we tweak the experiment such that each vocabulary list is learned by a different test subject, we introduce differences in capabilities between test subjects, making the evaluation of the list's efficiencies much more complicated to measure. To make up for these inconsistencies, we might boost the number of test subjects to the point where each list is learned by a statistically significant group of subjects, but this would cause a sharp increase in the cost of the experiment.

However, this theoretical test setup could be feasible using a non-human test subject: Using AI models and Explainable AI to analyze their interaction with language, we could not only evaluate, but even compile lists with drastically reduced costs. This approach is laid out in the following chapters.

3.3 Experimental Setup with AI Model

Chapter content: I view AI models as containers of language knowledge. We can perform studies on it as though it were human, gaining knowledge "from the viewpoint of an entity interacting with language"

This work is interested in finding vocabulary lists that language learners can use to gain linguistic competence in their chosen field in the least possible time. Chapter 3.1 has specified how with the help of a proxy task, we can define this efficiency. Chapter 3.2 has suggested a hypothetical experiment for testing the efficiency of a vocabulary list with human test subject. However, we have established that the experiment is not easy to set up consistently, as it cannot be repeated with the same test subject on different lists, and a large pool of test subjects either means lower comparability of the scores or significantly increased costs.

To circumvent this issue, this work proposes the following idea: To replace the human test subject in the experiment described in chapter 3.2 with an AI model, facilitating a consistent and economical setup.

The following paragraphs will lay out how, in such a setup, we can model the concepts from chapter 2.1.2 such as linguistic context and utility with technological tools.

AI as a Test Subject

In recent years, AI models such as ChatGPT [Insert citation: "ChatGPT"] or BERT [Insert citation: "BERT"] have become highly adept at fulfilling language-related tasks such as Language Modeling [Insert citation: "LLMs"], Named Entity Recognition and Sentiment Detection.

Such models possess as level of "understanding" that extends to the meaning of words: They can recognize that two words such as "building" and "apartment" are close to each other in meaning, even though the words are dissimilar on a character level (this paper makes no claims as to whether this understanding is real to that of a human, only that the models behaves as though it were). Thanks to this semantic language "understanding", AI models surpass the purely statistical approaches that were the paradigm of NLP until recently [Insert citation: "paper contrasting traditional with AI approaches"]. Thus, instead of a human performing a language exam to increases in linguistic understanding (*utility*), we could let an AI model perform an NLP task on a corpus instead.

If we run the model on a specific corpus and mask some of the words in the input, it is expected that the performance will decrease in comparison to the full input. However, presumably some words will have a larger impact on the performance than others. If we view removing words from the input as the equivalent of a human not knowing a word in a text, this performance differential can be a proxy metric for word utility and for human language ability.

Modified Experimental Setup with AI

With this new proxy metric for word utility in mind, we can propose a new experiment. To test the efficiency of a vocabulary list, we modify the setup from 3.2 as follows:

Let a pre-trained AI model perform its NLP task on a corpus. At the start, we mask all of the tokens in the input to simulate a language learner who is a complete beginner and thus knows no words in their target language. We then progressively unmask the words from the vocabulary list in the input, and each time run the AI on the corpus to check the updated (and presumably improved) performance.

This will yield a plot of performance over unmasked words which will tell us how quickly the performance improves with unmasked words, similar to figure 3.1. A quickly increasing performance will correspond to an efficient vocabulary list, whereas an inefficient list will create a plot with low initial gains in performance.

The modified experiment addresses the two issues proposed with the original setup: It is cheap when compared to using a human test subject and unlike the human, the AI model has no memory of previous tests, meaning we can run the experiment consistently every time.

Listing 1 shows pseudocode for the described setup:

Algorithm 1 List Efficiency Evaluation

Require: voc_list, corpus, model, p_aim
Initialize *scores* ← empty list
Set *test_interval* ← 100
for $i \leftarrow 0$ to length(*voc_list*) step *test_interval* **do**
    vocabulary ← *voc_list* until index i
    Initialize *line_scores* ← empty list
    for each *line* in *corpus* **do**
        line_with_only_known_words ← *line* without words not in *vocabulary*
        baseline_output ← model(*line*)
        new_output ← model(*line_with_only_known_words*)
        score ← similarity(*baseline_output*, *new_output*)
        Append *score* to *line_scores*
    end for
    if *avg_score* > *p_aim* **then**
        return -i
    end if
end for
▷ If the threshold score could not be reached with the list
return $-\infty$

Next, let us see how the components in this setup interact with each other to model the concepts introduces in chapter 2.1.2:

Influence of Components on Result

The evaluation of the vocabulary list efficiency in the experiment will depend on the following components:

- The AI model employed
- The input corpus
- The NLP task used

We have already addressed the role of the AI model in chapter 3.3. But the corpus and the NLP task performed by the model will also influence how quickly the AI model's performance increases with a given vocabulary list:

For example, when performing Sentiment Detection, words relating to emotion such as "hate", "like", "amazing" will matter more than if for an Information Retrieval task such as Temporal Tagging.

The corpus will have a more obvious influence on the performance, since corpora differ in many aspects such as formality, topic, and format of the text, all of which are components of a linguistic context.

Thus, by using diverse corpora, we can model various language contexts, and test word utility in those contexts. Once we have efficient vocabulary lists across those contexts, a language learners could choose one or several of these lists, based on what linguistic context is closest to their language learning goals.

Table 3.1 summarizes the correspondences between the concept and the technical component.

Abstract concept	**Technical Implementation**
Test subject	AI model
Language ability	Performance in NLP task
Language context	Corpus
Word utility	Impact of word on task performance

Table 3.1: Correspondence of abstract concepts to parts of implementation.

The next chapters will tackle the problem of how to not only test list efficiency, but make lists that approach an optimally efficient list.

3.4 Evaluating vs. Making Lists of Useful Vocabulary

The previous chapter describes an approach for evaluating the efficiency of vocabulary lists, which measures how quickly the list lets a learner improve their understanding in a specific context. While it can be seen as an improvement on the human setup because of its improved consistency and economy, it describes only the process of evaluation, not generation. Actually finding optimal vocabulary lists is theoretically possible by testing every possible vocabulary list with all words in the language.

In practice, however, this would require an enormous amount of computational resources, as can be seen on a simple example:

Let us suppose we wish to find the most efficient vocabulary list of 1,000 words for some context. To speed up list generation, we can make a preselection by allowing only the most frequent 20,000 words in the context as candidate members of this list. This would yield a total number of $P(20000, 1000) = \frac{20000!}{19000!}$ possible vocabulary lists, all of which would need to be tested for efficiency to find the most efficient one.

We can optimize the evaluation process, however, since these tests consist of running the proxy task with a gradually increasing vocabulary, and each vocabulary will be used in the evaluation for many vocabulary lists. For our example, a vocabulary is a set of words formed by taking the first n elements of the vocabulary list where $n \in [0, 1000]$. If we only consider the number of possible vocabularies with a cardinality between 0 and 1,000, we get $\sum_{k=0}^{1000} \binom{20000}{k} \approx \frac{1}{2} \times 2^{20000} = 2^{19999}$ runs of the proxy task. Clearly, this is still an unfeasible amount of computation.

It can thus be seen that finding optimal vocabulary list is much too expensive when performed with a brute-force approach. For this reason, the next chapter will introduce an alternative method for finding efficient lists of vocabulary, based on XAI as a tool which extracts information about the linguistic skills of AI models.

3.5 Generating Efficient Lists of Vocabulary

The previous chapters have shown the challenges in evaluating the efficiency of vocabulary lists with human test subjects, and suggested a repeatable approach utilizing AI models, NLP tasks and corpora to overcome these challenges. We have seen, however, that this approach does not suffice to generate efficient vocabulary. Therefore, to achieve cost-efficient list generation, this chapter advocates for adding a final component to our experimental setup, namely Explainable AI.

Explainable AI as a Tool of Analysis

The guiding question of this thesis is: *"Which words provide the most language understanding in a given linguistic context?"*. Chapter 3.3 has put forward an experimental setup in which the concepts of language understanding and context are simulated with technological components. By substituting these terms, the question is turned into: *"Which words provide the highest improvements in performance of an AI model performing an NLP task on a given corpus?"*.

Put like this, the question is very close to the questions that the field of Explainable AI seeks to answer. A typical question of Explainable AI would be: *"Why, for a given input, does the model arrive at its output?"*. [Insert citation: "definition of XAI. How to not make this seem like a direct quote?"] To be more precise, this is a question for a *local* explanation (see chapter 4.3). As mentioned in chapter 4.3, we will use Feature Attribution methods, which for an NLP task transform the question into: *"Which words in the input have the most influence on the model arriving at its output?"*

By answering this question on a statistically relevant sample of individual inputs from a corpus, we can find out which words are the most essential for the AI model to perform its task in the corpus overall. If the words found in this way are close to those that would provide the most utility to a human learner as well, they could be used to create very efficient vocabulary lists as well.

Once we have calculated the utilities of words, we only need to sort the words by their utility to arrive at a vocabulary list that should be close to optimally efficient. It must be noted that this is only an approximation of a maximally efficient list. This is because, theoretically, the utility of two words combined could be higher than the sum of their individual utilities. In other words, our approach ignores possible synergistic effects between words.

By adding XAI as a tool for analyzing the interaction of the AI model with the corpus, we finally have an experimental setup for generating efficient vocabulary lists. This is not a replacement of the approach described in chapter 3.3. Rather, this work will use the generation approach to make lists (chapter 4), and the evaluation approach for testing which method leads to the most efficient lists (chapter 5).

Algorithm 2 shows pseudocode for the described setup.

In the next chapter, we will tackle the implementation of this approach. We will present various choices for these components, and argue for some to be used over others, with the aim of making the utility estimated by the framework align as much as possible with utility to human learners.

Algorithm 2 Efficient List Generation

Require: corpus, model, xai

Initialize $line\_word\_utilities\_scores \leftarrow$ empty list
for each *line* in *corpus* **do**
    $word\_utilities\_for\_this\_line \leftarrow xai(model, line)$
    Append $word\_utilities\_for\_this\_line$ to $line\_word\_utilities\_scores$
end for
$corpus\_word\_utilities\_scores \leftarrow$ aggregate_word_utilities($line\_word\_utilities\_scores$)
$voc\_list \leftarrow$ words ordered by $corpus\_word\_utilities\_scores$
return voc_list

Chapter 4

Implementation

Chapter 3 has formalized the problem and put forward a novel framework for finding useful words, consisting of two major functionalities: Vocabulary list generation, and vocabulary list evaluation. The list evaluation approach, described in chapter 3.3), utilizes the performance of an AI model, corpus, proxy task as a proxy metric for language ability, in order to estimate how efficiently the vocabulary list may help a human language learner acquire competency.

The list generation approach, proposed in chapter 3.5, additionally uses Explainable AI as a tool for analyzing the interaction of the AI model with the corpus to compile vocabulary lists that approach maximal efficiency.

In this chapter, we will describe our implementation, mostly of the list generation approach. This is because the list evaluation method will be used in chapter 5 as one among several metrics for evaluating the efficiency of vocabulary lists generated by the implementation in this chapter. However, the evaluation with our approach will use the same components as the list generation approach, such as the chosen corpora, AI models, and NLP tasks.

ToDo: Write rest of introduction when chapter structure stands.

4.1 NLP Tasks

The NLP task and the corpus used for list generation

There are many NLP tasks we could choose from [Insert citation: "huggingface NLP task list or something"]. But not all are equally suitable. This chapter will first put forward criteria for selecting NLP tasks in chapter 4.1.1 for word utility evaluation. Afterwards, we use these criteria to select a few NLP tasks as appropriate in chapter 4.1.2. The NLP tasks we choose will dictate the format of our input data, i.e., the corpora we can use. Therefore, the selection of corpora will be discussed in the next chapter.

4.1.1 Desiderata

Chapter content: Support many languages Be as close to human need as possible -¿ task should show general language understanding data is easy to acquire -¿ data should be freely available for lots of contexts -¿ no need for manual labeling data

This section will put forward four criteria for selecting NLP tasks for word utility extraction: The availability of many multilingual AI models and corpora usable as input for the task, generality of the task and ease of evaluation. These criteria will be used in the next chapter to select concrete tasks for our implementation.

Model Availability in Many Languages

The goal of this work is to find approaches to find useful words for the purpose of language learning. Much research in Natural Language Processing is dedicated to improving NLP performance in English and other high-resource languages such as Spanish, French or Mandarin Chinese [Insert citation: "resource availability in various languages"]. This has the consequence that many AI models and other NLP methods achieve high levels of performance only in these languages, and many AI models are only available in English or a only small number of languages. However, there are over 7,000 languages in the world [Insert citation: "Ethnologue"], and for many of these there exist corpora, or online digital texts which could hypothetically be used as inputs for our word utility extraction approach. For this reason, our implementation strives to realize word utility extraction in **as many languages as possible**.

Corpus Availability in Many Languages

The second point of consideration for task selection is **how much data is available for performing the task**. Ideally, we would like tasks for which suitable corpora are freely available or can be trivially generated from available corpora. This is for the reason that, with a larger amount of usable data, we not only improve the accuracy of our approach, but also increase the diversity of input data. With diverse input data available, we have a larger amount of linguistic contexts for we can find utilities, and the context-specific language learning is a central motivation of this work.

Generality of Skill Required

Another important point to consider when selecting an NLP task is **how general the linguistic skills** are that the task requires: We use the performance in the NLP task as a proxy metric for the test subject's language ability. As such, we must ensure that task reflects a general level of semantic understanding, not only a narrow mechanistic skill that can be accomplished by using only a small part of the input.

Ease of Evaluation

To ensure we can measure the performance, we must also choose a task whose results can be easily compared with each other: Some tasks, such as text summarization, present a challenge for automatic evaluation: It is difficult to put a number on how similar two summaries of a text are. While evaluation measures, such as BLUE scores[Insert citation: "BLEU"], exist, it is questionable how well they capture the similarity between texts, because they do not recognize the semantic similarity of synonyms, and a different sentence structure will result in a low BLEU score even if the actual meaning of the sentences may be very close. It follows that if we have the freedom to choose NLP tasks whose results can be automatically evaluated with a fair degree of accuracy, we should choose them.

Summary

To summarize our desiderata for NLP tasks: Our implementation seeks to use NLP tasks for which both AI models and corpora exist in a large number of languages, to maximize accuracy and diversity of linguistic contexts which can be modeled. We prefer tasks that demonstrate general language understanding over tasks that only require a narrow skill set to perform or that are too technical, because general tasks are expected to align more with human linguistic skills. Finally, the task must be easily scorable, since the task score is the metric by which we gauge how useful words are.

The next section will present several tasks which fulfill the above criteria, and explain what consequences their selection has on the other components of vocabulary list generation. The choice of NLP tasks employed to test a XAI-based approach for word utility estimation is a crucial step: Since we are trying to estimate the utility a word a word has to language understanding, the NLP tasks should reflect language understanding as much as possible.

4.1.2 Task Selection

A good place to start looking for such tasks are those which are typically employed for pre-training NLP models: Pre-training tasks are used to first endow the AI model with a general understanding of the language, before using transfer learning to specialize it for a more specific downstream task. Such tasks must necessarily be general and require general language understanding, since training the model with them is supposed to provide a solid basis for a wide variety of NLP tasks. Another benefit of using pre-training tasks is that their training is unsupervised, meaning there is no need to manually label data. ToDo: Look at various pre-training tasks, preferably those used by state-of-the-art AI models ToDo: include free availability for AI models in their justification

4.1.3 NLP Pre-Training Tasks Used by State-of-the-Art AI Models

This section takes a look at the pretraining process of recent state-of-the-art LLM models which have made public their training process. Both the NLP tasks and the kind of data is considered.

GPT-4 [17]

Task: Language modeling (see next section).

Data: Not disclosed in detail, according to the original paper, the model was trained "using both publicly available data (such as internet data) and data licensed from third-party providers".

GPT-3 [4] GPT-3 is a model that does not rely on transfer learning to apply its linguistic understanding to new tasks; instead, it uses zero-shot and few-shot learning to perform tasks it was not specifically trained for.

Task: Language modeling (same as GPT-2 [19])

Data: Common Crawl, WebText2, Books1, Books2, Wikipedia ToDo: link sources?

LLama 3.3 [1]

Task: Meta did not make public the training process for Llama 3.3.

Data: "data from publicly available sources"

4.1.4 Tasks Considered

Next Sentence Prediction In this task, the AI model takes as input two sentences and predicts a probability for the second sentence being the successor of the first sentence in their source text. Advantages for this task for our purposes is that such a dataset is easy to generate, as it merely requires a corpus of sentences that follow from each other, which is easily obtained from Wikipedia articles, film subtitles, or any other continuous text.

Text summarization This task involves summarizing a given text, in other words, writing a shorter version of the input text while still conveying as much of the information from the original text as possible. Summarizing texts seems to require a high level of "understanding" of the text and would thus seem to be good choice for testing whether ablating certain words from the text would have detrimental effect on the model performance. Unfortunately, this task requires hand-labeled datasets and is thus not a good candidate if we aim to find approaches which can be implemented in many different languages, as there is a dearth in data in many of the less-studied languages of the world.

Masked language modeling (aka. "cloze task")

Causal language modeling (aka. Next token prediction)

Sentence order prediction

Sentence embeddings Sentence embeddings take the approach of transforming words into meaningful vectors and extend it to whole sentences. This "task" differs from the others in that we do not measure differences in performance when the input is perturbed; but rather a distance between the embedding vectors themselves. This justification for such an approach is that sentences whose meaning is very different should end up further apart from each other in the vector space once embedded. This brings several advantages: This approach can be performed on any corpus containing distinct sentences. These corpus does not have to be document-level, and sentences need not be consecutive. To make this a task on which XAI methods can be applied, we can define a distance from the original token

4.1.5 Sentence Embedding Methods

LASER [3]

BERT [20]

4.1.6 Data Required for Each NLP Task

The various NLP tasks employed require certain types of corpora to be employed properly:

xt sentence prediction Requires a corpus that contains consecutive sentences. Furthermore, NSP typically predicts whether two sentences follow each other in a document, not a dialogue (see the data on BERT training [11]). This excludes movie subtitles from the possible corpora for this task.

4.1.7 AI Models

- NSP-model ABC
- LLAMA?

ToDo: tests of performance tests of models used with corpora used (e.g., if NSP prediction model is reliable)

Attention as XAI can only be used on transformers

Tokenization (and thus selection of word candidates) is only independent on model use
As a direct consequence of this, other XAI mechanisms like attention as explanation are only useful for our purposes if the AI model uses a tokenization approach that somewhat corresponds to human notions of words. If a model uses tokenization approaches where a token is a combination of any three letters, any list obtained that tries to order the tokens by utility, while meaningful, will not be useful for human vocabulary learning. Note that in such cases, we can postprocess the data obtained, by merging the tokens to human-readable words and taking the average or maximum attention score of the AI model's tokens.

4.2 Corpora

4.2.1 Desiderata

For the purposes of this paper, it was desirable that the corpora used be:

- representative of what language learners strive for
- available in many languages
- for background corpora: Document-level
- freely accessible

4.2.2 Corpora Used

OpenSubtitles Parallel Corpus This set of corpora contains parallel corpora: Corpora which has text segments in one language aligned with the presumed translation of the segment in a second language. Its sentences are generated from subtitles from the popular subtitle sharing platform *OpenSubtitles* (https://www.opensubtitles.org/) and undergo various preprocessing and filtering steps as described in [15]. These include:

1. Enforcing universal UTF-8 character encoding.
2. Splitting and joining of sentences from their original subtitles blocks (the segments which appear on screen when watching the movie with its subtitle). One such block may contain multiple sentences, or only a partial one. There is thus a n-to-m-relationship between the blocks and sentences.
3. Checking and correcting possible spelling issues, especially ones arising from OCR (Optical character recognition) errors.
4. From available subtitles, identifying the subtitle pair which is most likely to be accurate in its alignments and free from errors such spelling, taking into account metadata such as user ratings of subtitles.

One advantageous aspect of this corpus is that is contains many sentences that are sequential, which means we can generate a Next Sentence Prediction dataset from it (add hedging here since not all lines in corpus are sequential and even within the same movies there may will be pauses in the subs). This corpus has been used to train machine translation models such as OPUS-MT [23], a freely available set of transformer models for translation, including between low-resource languages. ToDo: Not completely correct, the pipeline uses data from OPUS, not necessarily or specifically from OpenSubs While it is possible to reconstruct which movies the subtitle lines came from from information contained in the corpus, it is unfortunately not clear how these movies were selected in the first place.

Leipzig Wortschatz Corpora Available in x languages

But: data quality issues, methodology might be outdated

[6]

CCMatrix / NLLB

The full process, as illustrated by the authors, can be seen in figure 4.1 As of 2025, the latest version of the corpus (v2018) contains aligned subtitles of 62 languages between each other.

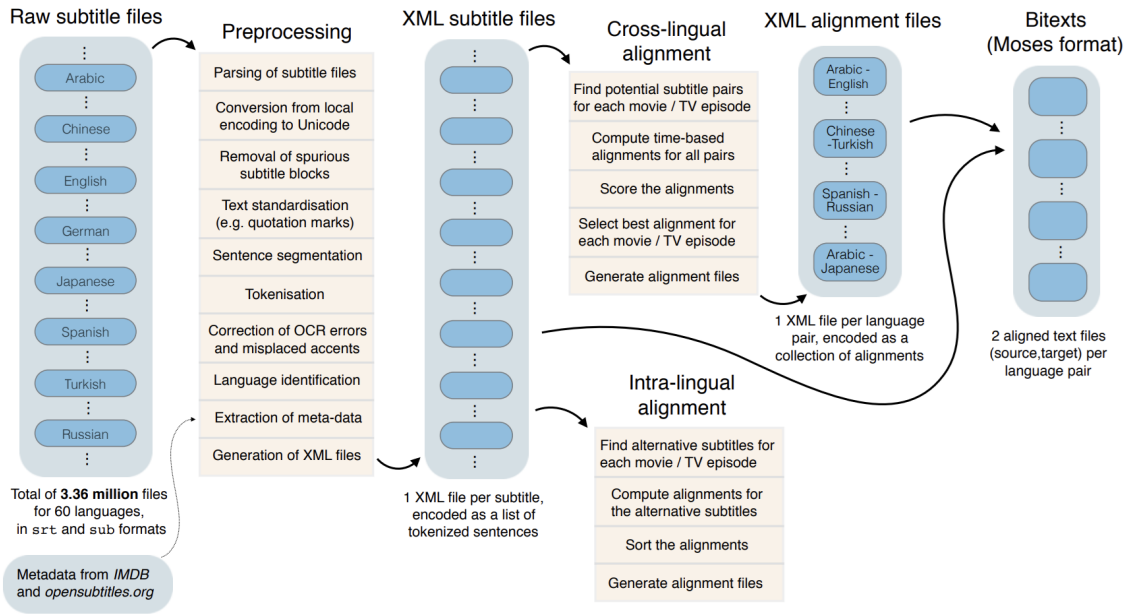

Figure 4.1: The pipeline producing the OpenSubtitles parallel corpus

4.3 XAI Methods

The XAI methods used in this work are the following:

- Attention as Explanation Advantages: Model only needs to be run once per sentence. Longer sentences do not lead to a much longer calculations Disadvantages: Justification as explanation controversial.
- Single Token Ablation

The following lays out how each of these methods works to achieve the goal of word utility estimation.

Performance difference of AI for NLP tasks Here, a Large Language Model (LLM) or a more specific language processing model is made to run NLP tasks such as text summarization, sentiment detection or question-answering. To find out which words help the AI model the most in performing its tasks, words are methodically omitted from texts and the AI's performance is recorded. This metric attempts to approximate utility by finding words which, when missing, cause the greatest performance loss in the NLP tasks. Evaluation metrics like Shapley values [26] may be used to measure the impact of missing words

Transformer attention The transformer architecture is based on a mechanism called *self-attention*. It allocates the neural network's processing to important parts of the input and thus provides some degree of explainability "out of the box".

Difference in internal vector representation for AI reading text This approach words similarly to the above involving an AI model, but instead of measuring the changes in the quality of its output, it measures how much changing the input to the model changes its the internal vector state: AI stores data in vector format, and when performing NLP tasks on texts, there is an internal vector representation. By using various distance metrics, it may be possible to find out which words have the greatest impact on the model's understanding of a text. Most of these approaches can be done both for individual words and word sequences (n-grams). While individual words are the easiest to examine, sometimes n-grams are insightful for finding sequences of words whose meaning is more than the sum of their parts (idioms and collocations) and which therefore must be learned in separately from their constituents (meaningful English n-grams include e.g. "kick the bucket", "such that", "such as").

This also raises the question of what is considered a "word". A phrase like "such as" can be considered two words if the definition of a word is simply "something separated by a space" or one word if the definition is "a phrase whose meaning cannot be arrived at trivially from knowing the definition of its parts". In Natural Language Processing, tokenizers break down texts into words, but they typically use the first definition for a word in the case of English. Many non-European language do not use spaces in their spelling (e.g. Japanese, Mandarin Chinese) or use spaces to separate a different unit of text (syllables in Vietnamese, sentences in Thai), making this definition of a word unpractical. In most languages, words can appear in various different

forms: Verbs in Spanish are conjugated according to the time and originator of an action, Nouns in German are declined depending on their number and grammatical case. This adds another variable for compiling word lists: Whether the list should consider any different combination of letters as a different word, or whether different forms of the same headword should be viewed as only one word.

4.3.1 Tokenizers

[explanation of why tokenizers are important, explain various possible definitions of "word"] [explain why morphosyntactically rich languages necessitate word splitting to some extend]

In some XAI models, we are free to choose any tokenizer we like. We can choose, for instance, to only use full words, or word parts in English. In input perturbation XAI approaches, we can choose to mask any part of the input with the help of tokenizers. For decomposition approaches, the model is not looked at as a black box but instead examined using model-specific methods such as attention or Layer-based Relevance Propagation. In such approaches, only the calculations made by the model itself are available for analysis, which means we are not free to choose our own word-splitting approach independent from the model. This is because these models are trained using a specific tokenizer in the preprocessing, and changing the preprocessing makes the model function incorrectly.

4.4 Data Pipeline

Chapter 5

Evaluation

5.1 Evaluation Measures

The various utility extraction approaches produce ranked lists as outputs. To compare these, we employ both quantitative and qualitative comparison approaches, described in the following chapters.

5.1.1 List Similarities

In order to compare the results provided by the various approaches, metrics are needed that can be consistently calculated across the different approaches. While a human may be able to qualitatively analyze lists and gain a rough idea of their similarity, computed metrics provide an instantaneous (if simplified) outlook on similarities. A metric shall be defined as a function which takes as parameters two word lists of equal length which are word lists ordered descendingly by supposed relevance, and outputs a real number giving either a distance or similarity between the lists.

Considerations of the choice of metric are:

Handling of lists with partial overlap. Metrics must be able to handle elements which occur in only one of the two lists. Thus, a metric which solely compares ranks of elements is not viable.

Start of lists is more impactful than end. Since the beginning of lists contains the words which are ranked as most important, changes at the top should impact the metric more than changes at the bottom. This includes (1) Equal differences in rank should be counted as more important if they occur further up the list. A word that is rank 1 in list A but rank 101 in list B says more about the similarity than if a word is rank 2000 in list A but rank 2100 in list B. Likewise, if a word is absent from list B, it implies a greater difference if that word is at rank 1 in list A than if it were at rank 1000.

Metrics Used

Sequential rank agreement (modified) [5]: This metric is based on the deviations of some subset of the lists in the upper ranks. It is important to note that this metric has an additional parameter "depth" which determines how many elements (from the top of the list) are considered. It is therefore more helpful to view its results at various depths. The original formula for this metric in the case of two lists is:

$$a_d := a \text{ from start to rank } d$$

$$S_d := a_d \cup b_d$$

$$SRA_d(a,b) := \lambda \cdot \frac{\sum_{x \in S_d} \sigma^2 \left((r_b(x)) - (r_a(x)) \right)}{|S_d|}$$

where λ is a normalization factor ensuring that $\max(SRA) = 1$. In its proposed form, this metric can only compare lists which contain the same set of unique

elements, just in different orders. In order to make it work on lists where this is not the case, one can set the "rank" of nonexisting elements to a value greater than the length of the lists, such as $2|a|$. Another drawback of the metric is that the standard deviation of two numbers does not depend on their absolute value, only their difference. However, to satisfy number 3 of the stated requirements, we can take the deviation of the logarithm of the ranks instead of the deviation of the ranks themselves, resulting in the formula

$$r'(x) := \begin{cases} \text{rank}_b(x) & \text{if } x \in b, \\ 2 \cdot |a| & \text{otherwise.} \end{cases}$$

$$SRA_d^{mod}(a,b) := \lambda \cdot \frac{\sum_{x \in S_d} \sigma^2 \left(\log(r'_b(x)) - \log(r'_a(x))\right)}{|S_d|}$$

For this modified version, λ can be calculated with:

$$\lambda = \frac{1}{SRA_d(a, a^*)},$$

where a^* is a list such that $a \cap a^* = \emptyset$.

Discounted Cumulative Gain : This formula outputs a value between 0 and 1, with 1 being given if both lists are identical, 0 when they have no elements in common, and values in between when there is partial overlap between elements and/or their order is different. $DCG_p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)} = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2(i+1)}$

$$rel_i := \begin{cases} \frac{1}{rank_b(el_i)+1} & \text{if } el_i \in b \\ 0 & \text{otherwise} \end{cases}$$

Metrics Rejected

Kendall rank correlation [10]: This metric is bounded between 0 and 1 and compares the ranks of the elements of two lists. However, it cannot handle elements that only occur in one of the two lists, and thus is not suitable for our purposes. It also does not distinguish between differences in the upper and lower parts of the lists.

Spearman's footrule [22]: Rejected for the same reasons as Kendall rank correlation.

Tests of Applicability

[Results of preliminary tests of various metrics on own lists such as rank switching ,replacement at bottom and top of list etc.]

5.2 Baselines

It may be useful to compare the lists generated by the various approaches with existing word lists from educational materials: Textbooks often feature chapters with word lists, or sentences which can be converted to word lists with a tokenizer. The purpose is to have a point of comparison, to see if generated lists agree with existing lists, and find reasons for differences.

5.2.1 Existing Lists

- Language learning textbooks
- Language learning applications
  - Duolingo: While Duolingo is the most popular language learning application as of 2024, it does not publish its word lists or course contents that is free of cost and easily convertible to a format that can be processed with NLP tools.
  - Rosetta Stone: Rosetta Stone publishes Course contents on its website. While the contents take the form of sentences, these can be converted to word lists by using a tokenizer on the contents and creating a list in the order in which they appear in the texts.

5.2.2 Existing Methods

To get an impression of how well the XAI-based methods perform, it will be informative to test established methods such

- Frequency: The frequency of the word in the context-specific corpus.
- Frequency without stopwords: The frequency of the word in the context-specific corpus, but with known stopwords filtered out.
- TF-IDF: The frequency of a word in the context-specific corpus, normalized against a background corpus.

5.2.3 LLM Prompts

In recent years, Large Language Models have become a popular tools for language learners to find new words to learn about specific areas. For this purpose, we run the following prompt to ChatGPT 3 for each context to get a sense of how well my method performs against this easy method. ToDo: Prompt and results

5.2.4 Sample text

ToDo: Show a sample paragraph with first n words from each list. Can be used to intuitively evaluate utilities

5.3 Results

Chapter content: Discussion/Analysis [Interpretation of the results] Could you combine approaches with each other or baselines to combine strength, what are the strengths/weakness of each approach

5.4 Discussion

Chapter 6

Outlook

- Sentence importance evaluation
- Capturing text as coherent unit instead of sentences which are independent from each other
- Active vs passive utility: Ideas for active utility estimation
- Compound splitting
- Lemmatization
- n-grams

Chapter 7

Appendix

7.1 Abbreviations

NSP Next sentence prediction

NLP Natural language processing

LLM Large language model

Bibliography

[1] Llama-models/models/llama3_3/MODEL_CARD.md at main · meta-llama/llama-models. https://github.com/meta-llama/llama-models/blob/main/models/llama3_3/MODEL_CARD.md.

[2] Alexander Clark, Chris Fox, and Shalom Lappin. The Handbook of Computational Linguistics and Natural Language Processing: Introduction. In *The Handbook of Computational Linguistics and Natural Language Processing*, pages 1–8. John Wiley & Sons, Ltd, 2010.

[3] Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the association for computational linguistics*, 7:597–610, 2019.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, and Amanda Askell. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] Claus Thorn Ekstrøm, Thomas Alexander Gerds, Andreas Kryger Jensen, and Kasper Brink-Jensen. Sequential rank agreement methods for comparison of ranked lists, August 2015.

[6] Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *LREC*, volume 29, pages 31–43, 2012.

[7] Xuehong (Stella) He and Aline Godfroid. Choosing Words to Teach: A Novel Method for Vocabulary Selection and Its Practical Application. *TESOL Quarterly*, 53(2):348–371, June 2019.

[8] S. Hunston. Corpus linguistics. In Keith Brown, editor, *Encyclopedia of Language & Linguistics (Second Edition)*, pages 234–248. Elsevier, Oxford, second edition edition, 2006.

[9] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. (unpublished), 3rd edition, 2025.

[10] M. G. KENDALL. A NEW MEASURE OF RANK CORRELATION. *Biometrika*, 30(1-2):81–93, June 1938.

[11] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2. Minneapolis, Minnesota, 2019.

[12] Tao Lei. *Interpretable Neural Models for Natural Language Processing.* Thesis, Massachusetts Institute of Technology, 2017.

[13] Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding Neural Networks through Representation Erasure, January 2017.

[14] Shaofeng Li, Phil Hiver, and Mostafa Papi. *The Routledge Handbook of Second Language Acquisition and Individual Differences.* Routledge, April 2022.

[15] Pierre Lison and Jörg Tiedemann. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, 2016.

[16] P. Nation. Vocabulary size, text coverage and word lists. *Vocabulary: Description, acquisition and pedagogy/acquisition and pedagogy*, pages 6–19, 1997.

[17] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine

McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. GPT-4 Technical Report, March 2024.

[18] Shahzad Qaiser and Ramsha Ali. Text mining: Use of TF-IDF to examine the relevance of words to documents. *International Journal of Computer Applications*, 181(1):25–29, 2018.

[19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[20] Nils Reimers and Iryna Gurevych. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation, October 2020.

[21] E. Rich. *Artificial Intelligence*. Artificial Intelligence. McGraw-Hill, 1983.

[22] Charles Spearman. Correlation calculated from faulty data. *British journal of psychology*, 3(3):271, 1910.

[23] Jörg Tiedemann and Santhosh Thottingal. OPUS-MT–building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, 2020.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. *Advances in neural information processing systems*, 30, 2017.

[25] Giulia Vilone and Luca Longo. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion*, 76:89–106, 2021.

[26] Rui Wang, Xiaoqian Wang, and David I. Inouye. Shapley Explanation Networks, April 2021.