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.

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 [9] 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 [12] 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]

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]

1.3 Outline of Work

Chapter 2

Background

2.1 State of the Art

2.2 Natural Language Processing

Chapter 3

Method

3.1 Overview

It is evident that recent tools based on Artificial Intelligence are much better-equipped for many tasks in the realm of Natural Language Processing than rule-based tools. At the same time, it is less clear how exactly these networks achieve the results they do. Fortunately, there is a branch in the field of Artificial Intelligence called Explainable Artificial Intelligence which aims at explaining the outputs of AI models. Thus, it may be possible to harness the evident intelligence off AI models to simulate a human, to find out which words have the largest impact on the model's performance.

3.1.1 Word extraction methods

The approaches for approximating utility with an automatically computable metric which this works aims to compare include:

Traditional

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 corpus 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.

AI-simulated learner

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 [17] 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.

Key technologies employed include therefore:

- Tokenizers
- Lemmatizers
- Translators
- AI models to perform NLP tasks

3.2 Components of XAI word extraction

Components are

- XAI method used
- Tokenizer
- (Pretrained) AI model used
- NLP task

It is readily seen that these components are not independent of each other. Some completely determine the choice of another, while others limit the selection of the other components. [describe dependencies between components] Task → model → tokenizer → words.

must investigate comparability of results later.

(also corpus → task)

3.2.1 Preliminary investigations of integrity

The core idea this paper is to evaluate word utility by investigating a function (in the form of an AI model) that presumably represents some level of understanding of the language and checking which inputs (words) have the biggest influence on either the input or the model's internal state. To ensure the function does possess this understanding, it is necessary to first ensure that the output of the model corresponds reliably to the ground truth, in other words, to tests the performance of the model on the specific data that will later be used to investigate the model itself.

Note: A lot of prelim. test results, for example, first tests of NSP model on opensubs sentences show low reliability. Might also have to move this section to end (or in "results" chapter?)

3.2.2 Formal problem statement for XAI word extraction

Givens:

- A set W of w candidate words: $|W| = w$.
- A corpus C containing lines/sentences in the target language.
- A function f indicating the performance at the chosen task when given the subset of W

$$\begin{aligned} f : 2^W &\to \mathbb{R} \\ f(K) &\mapsto p \end{aligned}$$

- An integer k denoting the desired cardinality of the (smaller) subset of words to learn.

Find

$$\arg\max_{K} f(K)$$

$$K \subset W$$
$$|K| = k$$
$$k < |W|$$

In practice, it is often not feasible to calculate calculate f for every possible subset K, necessitating the use of approximations.

3.3 NLP tasks

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. 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. Note: Look at various pre-training tasks, preferably those used by state-of-the-art AI models

3.3.1 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 [10]

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 [3] 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 [13])

Data: Common Crawl, WebText2, Books1, Books2, Wikipedia Note: 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"

3.3.2 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

3.3.3 Sentence embedding methods

LASER [2]

BERT [14]

3.3.4 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 [7]). This excludes movie subtitles from the possible corpora for this task.

3.4 XAI methods

- 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

3.5 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.

3.6 AI models

- NSP-model ABC
- LLAMA?

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

3.7 Interdependencies between components used

While the components described above can mostly be used in any combination, there are some important restrictions to keep in mind:

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.

3.8 Corpora

3.8.1 Selection criteria

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

3.8.2 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 [8]. 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 [16], a freely available set of transformer models for translation, including between low-resource languages. Note: 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

[5]

CCMatrix / NLLB

The full process, as illustrated by the authors, can be seen in figure 3.1 As of 2025, the latest version of the corpus (v2018) contains aligned subtitles of 62 languages between each other.

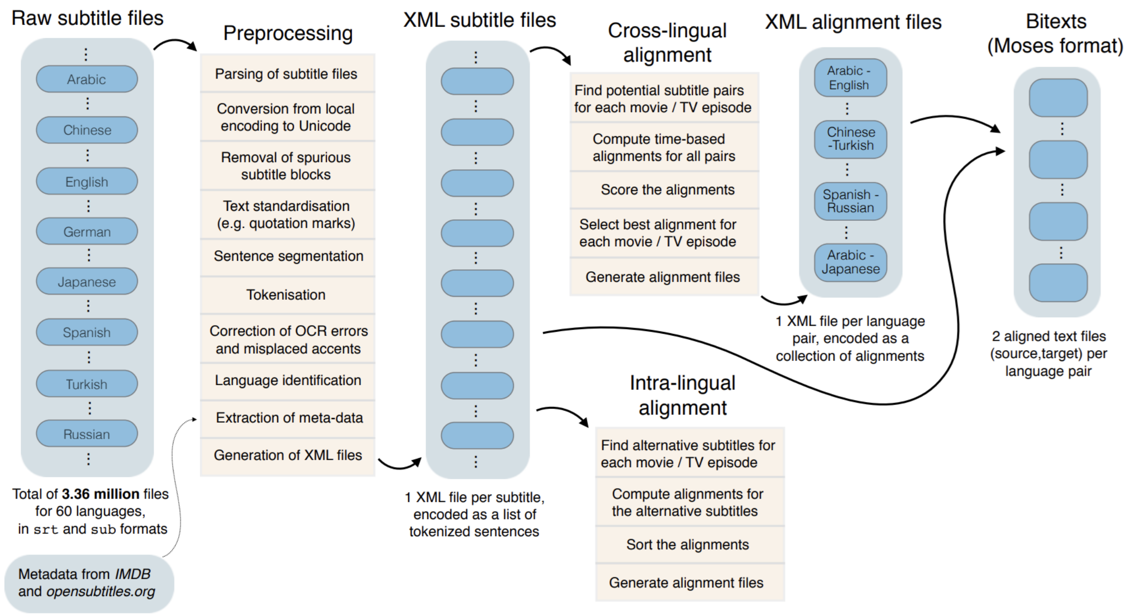

Figure 3.1: The pipeline producing the OpenSubtitles parallel corpus

3.9 Data Augmentation

Data Augmentation was not used to gain additional data. While in recent years data augmentation methods have become popular for training AI models in NLP, most of these would have either no or a detrimental effect on the methods employed in this paper: Some of these methods include [11]:

Character level Introducing character swaps to data is a method used to train the model to noise in the data, but in our case, would only add noise to the results.

Word level There exist techniques to switch words for synonyms or swap words in the sentences to create noise. As with noise on the word level, adding words in inappropriate places is undesired for our use case. Swapping words for synonyms would also prove detrimental, as this would skew the statistics away from the natural word distribution found in the human-generated source texts.

Higher level techniques suffer from the same issues. For this reason, the data is left in its "natural state" for our purposes.

Chapter 4

Implementation

4.1 Data pipeline

Chapter 5

Evaluation

5.1 Evaluation measures

The various utility extraction approaches produce ranked lists as outputs. To compare these, I 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) [4]: 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

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

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

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 [6]: 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 [15]: 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 Existing points of comparison

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.

- 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.

5.3 Results

5.4 Discussion

Chapter 6

Outlook

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] 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.

[3] 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.

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

[5] 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.

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

[7] 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.

[8] 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.

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

[10] 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.

[11] Lucas Francisco Amaral Orosco Pellicer, Taynan Maier Ferreira, and Anna Helena Reali Costa. Data augmentation techniques in natural language processing. *Applied Soft Computing*, 132:109803, January 2023.

[12] 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.

[13] 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.

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

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

[16] 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.

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