XRDict: XLM-R Cross-Lingual Reverse Dictionary

Armando Fortes
2021403383
Tsinghua University

David Pissarra
2021403381
Tsinghua University

Gabriele Oliaro
2021280352
Tsinghua University

Abstract

A reverse dictionary is a practical tool which allows you to find the words that best represent a given description of a concept. Unlike regular dictionaries, which have been around for a long time, reverse dictionaries were impractical before the advent of computer science, as there are many phrases which could describe a concept, and each such phrase can contain multiple words, and it would be very difficult to collect all entries in a paper-based reverse dictionary, and equally difficult to sort them. Recent developments in deep learning have made it possible for the users to simply type in their own words the description of a concept they are looking for, and let the machine learning model search for the best match. Previous works have already achieved impressive results, however, many improvements can still be made to existing NLP-based reverse dictionary models. In this research project, we aim to build our own neural-network based model and contribute to the research community by investigating ways to improve or generalize the existent state-of-the-art models. *Source code:* `https://github.com/davidpissarra/XRDict`

1 Introduction

Unlike regular dictionaries, a reverse dictionary maps given descriptions and definitions to the desired target words. This process is also know as onomasiological search (Sierra, 2000).

Reverse dictionaries are an extremely useful tool for people who write considerable amounts, such as writers, translators and researchers. Additionally, language learners may also find these type of dictionaries a facilitative tool, since it helps them express their thoughts better, by overcoming the limitations of their vocabulary. Moreover, they are specially useful to help solve the "tip of the tongue" phenomenon (Brown and McNeill, 1966), which almost everyone has already experienced before.

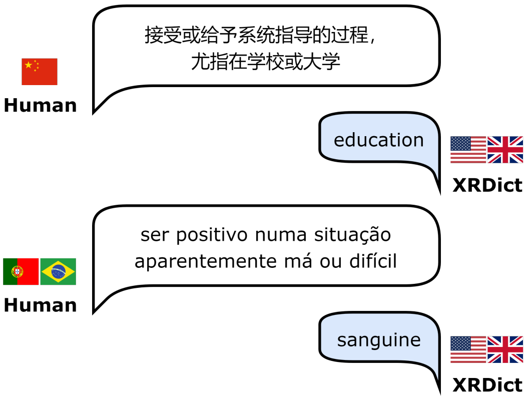
接受或给予系统指导的过程，
尤指在学校或大学
Human
education
XRDict
ser positivo numa situação
aparentemente má ou difícil
Human
sanguine
XRDict

Figure 1: **Cross-Lingual Reverse Dictionary Intuition.** An example of the usage of our proposed cross-lingual reverse dictionary model.

From a natural language processing perspective, building a reverse dictionary is an interesting application, as it consists of creating a model that is able to learn how a concept can be represented in lexical form (using a single word) or as a phrase, and be able to map instances of the second form to the first (Hill et al., 2016).

Recent work such as (Qi et al., 2020) implement a reverse dictionary using a transformer architecture, and achieve far better results than previous implementation based on RNN or LSTM techniques. However, many improvements can still be made to existing reverse dictionary models, and the field is still an area of active research. In this research project, we aim to build our own neural-network based reverse dictionary and compare its performance to previous work using datasets such as "The online plain text English dictionary" (OPTED) or WordNet (Miller, 1995).

The current state-of-the-art reverse dictionary is WantWords (Qi et al., 2020), which uses BERT (Devlin et al., 2018) as its pre-trained language model. In our project, compared to WantWords, we used a new architecture, performed new exper-

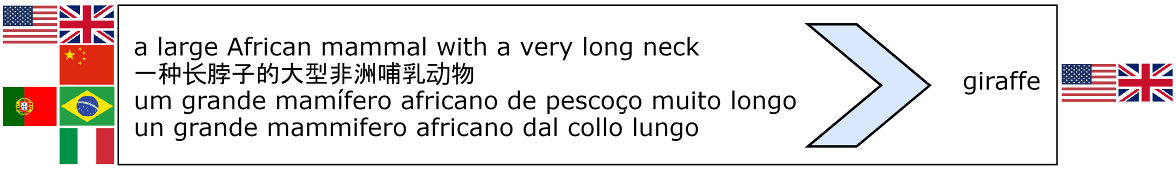
a large African mammal with a very long neck
一种长脖子的大型非洲哺乳动物
um grande mamífero africano de pescoço muito longo
un grande mammifero africano dal collo lungo
giraffe

Figure 2: **Cross-Lingual Reverse Dictionary Example.** In this example, we present the general usage of the cross-lingual reverse dictionary. Given word definitions in some of the supported languages, our model intend to output the corresponding word.

iments and, used newer datasets. We considered several newer pre-trained models such as DistilBert (Sanh et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2019), XLNet (Yang et al., 2019) or GPT-2 (Radford et al., 2019), which have been invented and open-sourced since the release of WantWords. Eventually, we settled for the multilingual XLM-RoBERTa (XLM-R) (Conneau et al., 2019) by Meta. Our reverse dictionary aims to be more general than than WantWords, and support cross-language lookups natively. In other words, users can type the definition of a concept in any language, and obtain a list of English target words that best match the concept. We picked English as the language to be used for the target words, because it's the most widely-used *lingua franca*; however, it would be possible to swap English with any other language of interest. Even simpler, one could couple the reverse dictionary with a bilingual dictionary, and translate the target words from English to another language.

In the rest of this report, we discuss the related works on reverse dictionaries and cross-lingual language understanding (section 2); then we introduce the details of the methodology used to implement XR-Dict (section 3) and we describe our evaluation efforts (section 4). Finally, we present our conclusion (section 5).

2 Related Work

Since the breakthrough of neural networks, many different solutions emerged in order to solve the "tip of the tongue" problem. Recurrent neural networks (RNNs) started by showing some efficiency in this sentence-word matching task (Hill et al., 2015), by extracting continuous knowledge from the input queries, which in turn can match them with the most similar word embeddings. However, due to the shortcomings of RNNs, alternative approaches were introduced such as graph-based reverse dictionaries (Thorat and Choudhari, 2016). By using a distance-based similarity measure, computed on a graph, the similarity between the input definition and vocabulary words can be quantified.

More recently, in an attempt to solve previous models' issues, such as highly variable input queries and low-frequency target words, a multichannel reverse dictionary was proposed (Zhang et al., 2020). This model is based on a bidirectional LSTM (Long Short-Term Memory), implementing an attention mechanism. Additionally, four characteristic predictors were added to the model in order to predict the part-of-speech (POS), morphemes, word category and sememes of the target word, based on the input query. Moreover, on top of the aforementioned model, an open-source online reverse dictionary system called *WantWords* was created (Qi et al., 2020). The core module of this system is the multi-channel reverse dictionary, but, instead of using the bidirectional LSTM, the authors decided to employ BERT (Devlin et al., 2018) as the sentence encoder. In addition, this system also supports a cross-lingual mode, which consists on translating the input query to the target language using an external translation model, followed by executing the same procedure with the monolingual mode.

Our efforts to create a truly cross-lingual reverse dictionary (without having to translate the definitions provided by the user into the language of the target words before feeding it to the dictionary) were enabled by previous work in cross-lingual language understanding, and by the availability of cross-lingual pre-trained models. The current state of the art is the XLM-R model (Conneau et al., 2019), which we use in our dictionary. The strength of XLM-R comes not only from its architecture, but also from having been trained on the Common Crawl (Foundation) dataset, a very large (over 2TB) and highly multilingual one. In addition,

Vocabulary Size	Word
1.000	c r os s walk
5.000	cr oss walk
10.000	cross walk
50.000	crosswalk

Table 1: **Example of the BPE tokenization applied to NLP tasks.** Tokens are merged recursively, prioritizing those with the highest score. As the vocabulary size increases, rare words are completely formed more easily.

XLM-R uses the techniques from (Wenzek et al., 2019) to generate a cleaned version of the Common Crawl dataset that optimizes the performance (improving it by two orders of magnitude compared to the previous state-of-the-art) of the model with those languages for which less training resources are available.

3 Methodology

We built our reverse dictionary model using Pytorch. Our model takes as input a sentence, describing a concept in one of the supported languages. The output is a list of words that best match that concept in a language of choice. The data processing (see section 3.1) and BPE tokenization (see section 3.1.1) are performed in the same way for the training dataset as well as the sentences used for prediction and testing. In addition, we describe our selected sentence encoder i.e., the XLM-R (see section 3.2), as well as, the overall details of the architecture (see section 3.4).

3.1 Data Pre-Processing

We start by pre-processing our dataset, removing punctuation and converting words to unicode characters, creating simple tokens. We further finalize our data pre-processing, by applying BPE tokenization (see section 3.1.1).

3.1.1 BPE Tokenization

Throughout different Natural Language Processing tasks, one of the first things done when feeding text to some model is to split it into individual words. But in fact, since we are approaching a cross-lingual problem, these individual words have specific meanings to each language, being some of these poorly defined for some languages. Additionally, simple tokenizers would generate a very large vocabulary, due to regarding each unique word as a different token. This makes designing a tokenizer relatively complex, when it takes to consider many different language specific rules, which ends up being slow. Also, besides having a large vocabulary, which is considered to be a problem, most of the words will not be covered by the entire vocabulary, due to the variety of languages.

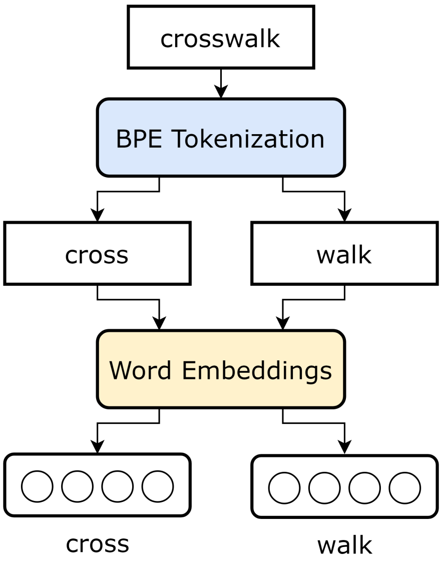

Figure 3: **BPE Tokenization to Word Embedding.** Words are splitted according the BPE codes, resulting in tokens. These tokens are associated with its own word embedding.

To tackle the mentioned issues, Sennrich et al. (2015) proposed a word segmentation model based on *byte pair encoding* compression algorithm (BPE). This approach intends to solve these issues by splitting the text into frequent n-grams, reducing the overall size of the vocabulary. Subsequently, unseen words could also benefit from this technique, since rare words can be built from n-grams. On the other hand, common words will be entirely built as a n-gram since will have significant frequency. Therefore, we apply this algorithm to the processed text before feeding it to our model, where each resulting token will be associated with its word embedding.

3.2 XLM-R

Since the breakthrough of BERT (Devlin et al., 2018), some multilingual masked language models (MLM) such as, the multilingual BERT approach of Devlin et al. (2018) and XLM (Lample and Conneau, 2019) have pushed the state-of-the-art on cross-lingual understanding tasks. Nevertheless, these tend to struggle when learning useful

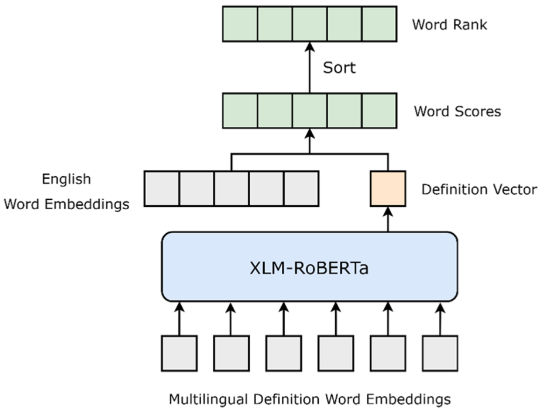

Figure 4: **XRDict Architecture.** Our model architecture is composed by a pre-trained cross-lingual model (XML-RoBERTa) as the sentence encoder. Moreover, a classifier will be responsible for mapping the sentence vectors to the target embedding space, as well as computing the respective confidence scores.

representations for low-resource languages. Thus, XLM-R (Conneau et al., 2019) improves on previous approaches by training RoBERTa (Liu et al., 2019) on a very large dataset that had been cleaned and filtered.

We therefore propose to use XLM-R as the sentence encoder of our model. This way, preprocessed text will be fed to the XLM-R, which in turn will output a definition vector, carrying the intrinsic meaning of the given sentences.

3.3 Word Embeddings

After performing the data processing and BPE tokenization on each sentence, each BPE token will be associated with its own word embedding, further feed to the sentence encoder. Even if some word is divided into several tokens, it will be composed by the tokens' word embeddings. But, it turns out that, we distinguish the definition word embeddings from the target word embeddings, since definitions are produced in many languages, and the selected target language is english. So, not only aiming to reduce the dimensionality, this is the reason why it is so important to map the definition vector to the target embedding space (see section 3.4).

3.4 Overall Architecture

The overall architecture of our proposed model is based on the sentence classification model, i.e., composed by a sentence encoder followed by a classifier. As mentioned in the previous subsection, we choose XLM-R as the sentence encoder, which will encode input queries into vectors. Therefore, this transformer-based pre-trained model, will capture the importance of each word on the representation of the sentence.

Formally, every input query is first encoded into a sentence vector by XLM-R. We perform this operation, by fetching the special classification token's (`[CLS]`) final hidden state, which in fact is the first vector of the XLM-R output sequence. The special classification token was firstly introduced in BERT (Devlin et al., 2018), but has been widely used in recent transformer-based pre-trained models in order to aggregate sequence representation for classification tasks. Subsequently, the resulting sentence vector is mapped into the space of the target word embeddings by a single-layer perceptron:

$$\mathbf{e}_d = \mathbf{W}\mathbf{v}_{\texttt{[CLS]}} + \mathbf{b}, \tag{1}$$

where $\mathbf{e}_d$ is regarded as the definition vector in the embedding space, $\mathbf{v}_{\texttt{[CLS]}}$ is the resulting XLM-R sentence representation vector, $\mathbf{W}$ is the single-layer perceptron weigth matrix, and $\mathbf{b}$ a bias vector.

Thereupon, the confidence score of each word, regarding the definition vector, can be computed as a simple dot product:

$$\mathbf{score}_{d,w} = \mathbf{e}_d \cdot \mathbf{e}_w, \tag{2}$$

where $\mathbf{e}_w$ is the target embedding vector of the word w.

Finally, in an attempt to retrieve the highest scored target words, we sort the entire set of target word scores:

$$\mathbf{top}_k = \mathbf{sorted}(\mathbf{score}_{d,\mathbf{W}}, k), \tag{3}$$

where **sorted** is a sorting function (descending order), $\mathbf{score}_{d,\mathbf{W}}$ is the entire set of target word scores, and k the number of desired highest scored target words.

4 Experiments

The objective of this section is to conduct a detailed quantitative analysis of the performance of our proposed cross-lingual reverse dictionary model. We carry out experiments on a new multilingual dataset which supports mapping English, Chinese, Portuguese and Italian descriptions to their respective English target concepts.

Model	*median rank*	*accuracy@1/10/100*	*rank variance*
XRDict	13	.23/.49/.74	253

Table 2: **Overall XRDict Performance.** We present the results for the cross-lingual reverse dictionary task. We regard the accuracy of target words appearing in the top 1/10/100 word ranks prediction, median rank and rank variance.

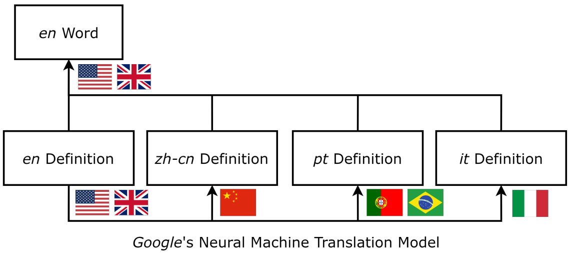

Figure 5: **Data Collection Procedure.** An overview of our data collection pipeline, where we perform definition translation using Google's Neural Machine Translation model to Chinese, Portuguese and Italian, while maintaining the previous word-definition pair correspondence from the English dictionary dataset.

4.1 Dataset

We create a new *many-to-one* multilingual reverse dictionary definition dataset, where English, Chinese, Portuguese and Italian descriptions are mapped to English target words. Building upon the English dictionary definition dataset created by Hill et al. (2016), we randomly subsample around $250,000$ word-definition pairs and perform definition translation using Google's Neural Machine Translation model (Wu et al., 2016) for each of the aforementioned supported languages. Accordingly, the dataset contains about $1,000,000$ word-definition pairs in total, with respect to a source BPE-tokenized vocabulary size of $200,000$ a target vocabulary size of 44,553. This dataset was then split into training, validation and testing sets, with the respective proportions of 70%, 10% and 20%.

4.2 Experimental Details

In this section, we present some of the settings and insights throughout the model experiments.

Hyperparameters and Training We fine-tune our model from a publicy available XLM-R model provided by `MetaAI`[1], pre-trained on the Masked Language Model (MLM) task on 17 different languages. In addition, the BPE codes and vocabulary are also provided, in order to proceed with the BPE tokenization. The vocabulary size of the given pre-trained model is 200,000, and the BPE token input embeddings have a dimension of 1280 (definition BPE token embeddings are fixed during training). Moreover, we use the 300-dimensional word embeddings pre-trained on GoogleNews with `word2vec`[2] for our target words, and these word embeddings are also fixed during training. For the XLM-R fine-tuning process, we adopt Adam as the optimizer with initial learning rate of $5e-6$, a batch size of 32, and the model is trained for 30 epochs with early stopping settings to prevent overfitting.

Zero-shot Cross-lingual Transfer Additionally, after fine-tuning our model, we conclude that the model successfully could infer other languages, learned before on the pre-training phase, without even using those throughout the fine-tuning process, since XLM-R is surprisingly good at zero-shot cross-lingual model transfer.

Evaluation Metrics Based on previous research, we use three evaluation metrics: the median rank of target words (lower is better), the accuracy of target words appearing in the top 1/10/100 word ranks prediction (higher is better), and the rank variance of the target words (lower is better).

4.3 Experimental Results

We present the results of our model in Tab. 2. The performance of XRDict is evaluated using the accuracy of target words appearing in the top 1/10/100 word ranks prediction, median rank and rank variance. To conclude, our model showed competitive results when compared to other reverse dictionaries results, despite these being based on the monolingual approach.

5 Conclusion and Future Work

In this project, we proposed a cross-lingual reverse dictionary model based on XLM-R, which is currently trained on four definition source languages: English, Chinese, Portuguese and Italian; and one

[1] https://github.com/facebookresearch/XLM

[2] https://code.google.com/archive/p/word2vec/

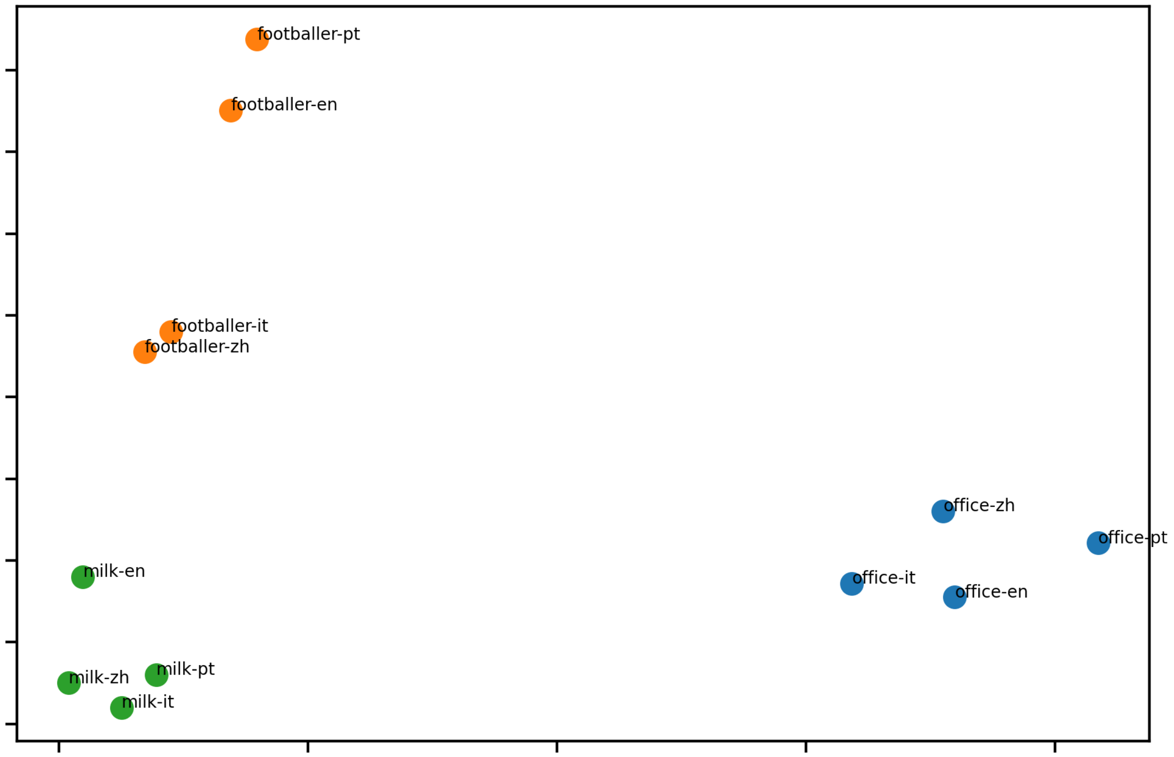

Figure 6: **XLM-RoBERTa Definition Vector Space Representation Using PCA.** We plotted the 1280-dimensional definition vectors for some word definition in different languages (*zh*, *en*, *pt*, *it*). In order to visualize these, we reduced the vector dimensionality using Principal Component Analysis (PCA).

target language: English. However, our model analogously supports the addition of more source languages either by collecting and training such definitions as described in this report (see section 4), or by simply inferring them, since XLM-R will already map such definitions similarly to their counterparts from the supported languages in the definition vector space.

Our model performed well in the quantitative analysis done, since the results presented are very competitive against current state-of-the-art models, even in the monolingual reverse dictionary task. Nevertheless, due to the computational complexity of our model and training process, and due to time constraints, other insightful experiments that could also have been done are unfortunately not presented. Prominent examples of desired experiments for future work would be: reproduction of previous state-of-the-art works for head-to-head performance comparison with our model, more complementary data assessment based on human-written descriptions instead of just dictionary-based definitions. We also would like to create a more robust dataset, without having to translate definitions to other languages. Thus, we also could test our model with unseen and already seen definitions, jointly with human-like descriptions that could bring value to the sentence encoding phase.

In the future, we would like to upgrade our model from a *many-to-one* end-to-end architecture to a *many-to-many* one, where the target embedding space would also be multilingual. Although such architecture would probably require a more complex classifier architecture, we believe the work we present lays a useful foundation for such future works.

References

Roger Brown and David McNeill. 1966. The “tip of the tongue” phenomenon. *Journal of verbal learning and verbal behavior*, 5(4):325–337.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

The Common Crawl Foundation. The Common Crawl corpus. https://commoncrawl.org/the-data/get-started/. Accessed: 2022-06-26.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2015. Learning to understand phrases by embedding the dictionary. *CoRR*, abs/1504.00548.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to understand phrases by embedding the dictionary. *TACL*, 4:17–30.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

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

Fanchao Qi, Lei Zhang, Yanhui Yang, Zhiyuan Liu, and Maosong Sun. 2020. Wantwords: An open-source online reverse dictionary system. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–181.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Gerardo Sierra. 2000. The onomasiological dictionary: a gap in lexicography. In *Proceedings of the ninth Euralex international congress*.

Sushrut Thorat and Varad Choudhari. 2016. Implementing a reverse dictionary, based on word definitions, using a node-graph architecture. In *Proceedings of COLING*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2019. Ccnet: Extracting high quality monolingual datasets from web crawl data. *CoRR*, abs/1911.00359.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding.

Lei Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2020. Multi-channel reverse dictionary model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):312–319.