# word2word: A Collection of Bilingual Lexicons for 3,564 Language Pairs

# Yo Joong Choe\*, Kyubyong Park\*, Dongwoo Kim\*

Kakao Brain

20, Pangyoyeok-ro 241, Bundang-gu, Seongnam-si, Gyeonggi-do, Korea {yj.choe, kyubyong.park, dongwoo.kim}@kakaobrain.com

#### Abstract

We present word2word, a publicly available dataset and an open-source Python package for cross-lingual word translations extracted from sentence-level parallel corpora. Our dataset provides top-k word translations in 3,564 (directed) language pairs across 62 languages in OpenSubtitles2018 (Lison et al., 2018). To obtain this dataset, we use a count-based bilingual lexicon extraction model based on the observation that not only source and target words but also source words themselves can be highly correlated. We illustrate that the resulting bilingual lexicons have high coverage and attain competitive translation quality for several language pairs. We wrap our dataset and model in an easy-to-use Python library, which supports downloading and retrieving top-k word translations in any of the supported language pairs as well as computing top-k word translations for custom parallel corpora.

Keywords: bilingual lexicon, word translation, Python toolkit

# 1. Introduction

Bilingual lexicons (Fung, 1998) are valuable resources for cross-lingual tasks, including low-resource machine translation (Ramesh and Sankaranarayanan, 2018; Gū et al., 2019) and cross-lingual word embeddings (Ruder et al., 2017). However, it is often difficult to find a large enough set of bilingual lexicons that is freely and readily available across various language pairs (Levy et al., 2017). For example, standard bilingual dictionaries like Wiktionary¹ often do not explicitly provide word correspondences but refers or redirects to the query word's dictionary form:

Query: travaillé (French for 'worked')
 Result: (verb) past principle of travailler 'work'

• Query: 먹었다 (Korean for 'ate') Result: redirects to 먹다 'eat'

Not only does this make it tedious to find word-level correspondences across many query words, this is particularly problematic when we try to find word correspondences for languages where some dictionary forms are rarely used in ordinary discourse, such as the case of 먹다 in the Korean language.

While the task of bilingual lexicon extraction (BLE) has been popular in both early and recent literature, spanning from count-based approaches (Fung, 1998; Vulić and Moens, 2013; Liu et al., 2013) to using cross-lingual word embeddings (Ruder et al., 2017; Mikolov et al., 2013a; Gouws et al., 2015; Conneau et al., 2017; Levy et al., 2017; Artetxe et al., 2018; Artetxe et al., 2019), few were focused on building high-coverage bilingual lexicons across many language pairs, possibly including non-Indo-European languages. In fact, many of the recent studies and their accompanying packages (Conneau et al., 2017; Artetxe et al., 2018; Glavaš et al., 2019) aim at evaluating cross-lingual word embeddings, so that they involve at most 10-100s of language pairs and 1-5K words for each pair.

# Languages	62
# Language Pairs	3,564
Avg. Lexicon Size	127,023
Avg. # Translations Per Word	8.8

Table 1: Overview of the word2word dataset.

Motivated by the lack of publicly available and high-coverage bilingual lexicons across diverse languages, we present *word2word*, a large collection of bilingual lexicons for 3,564 language pairs across 62 languages that is wrapped around an open-source and easy-to-use Python interface. We extract top-*k* bilingual word correspondences from all parallel corpora provided by OpenSubtitles2018<sup>2</sup> (Lison et al., 2018), using a count-based model that takes into account both monolingual and cross-lingual co-occurrences. The package also provides interface for obtaining bilingual lexicons for custom parallel corpora in any other language pairs and domains not covered by Open-Subtitles2018.

# 2. The word2word Dataset

### 2.1. Data Statistics

The word2word dataset spans across 3,564 directed language pairs between 62 languages in the OpenSubtitles2018 dataset, a collection of translated movie subtitles extracted from OpenSubtitles.org<sup>3</sup>. By design, our methodology covers 100% of words present in the source sentences, making the lexicon size much larger than existing bilingual dictionaries. The lexicon also contains up to top-10 word translations in the target language. We provide an overview of the entire dataset in Table 1.

In Table 2, we provide summary statistics for bilingual lexicons between English and some of the major languages (both European and non-European). For each pair, the lexicon size ranges from 76.2K (English-Russian) to

<sup>\*</sup>Equal contribution.

https://en.wiktionary.org

<sup>2</sup>http://opus.nlpl.eu/OpenSubtitles-v2018.php

http://www.opensubtitles.org/

Language Pair	Lexicon Size	# Unique Translations	Avg. # Translations Per Word	# Sentences Used
Arabic-English	335.5K	86.0K	9.7	29.8M
English-Arabic	97.6K	191.6K	9.5	29.8M
S.Chinese-English	214.0K	87.0K	9.5	11.2M
English-S.Chinese	101.6K	139.1K	9.4	11.2M
T.Chinese-English	201.7K	72.5K	9.5	4.8M
English-T.Chinese	85.8K	119.7K	9.2	4.8M
French-English	92.1K	59.1K	9.8	41.8M
English-French	72.1K	71.4K	9.7	41.8M
Italian-English	111.5K	63.9K	9.7	35.2M
German-English	127.0K	64.8K	9.7	22.5M
English-German	73.6K	95.9K	9.6	22.5M
English-Italian	75.4K	83.9K	9.6	35.2M
Japanese-English	83.3K	75.2K	9.2	2.1M
English-Japanese	102.1K	63.8K	9.3	2.1M
Korean-English	87.2K	75.8K	9.3	1.4M
English-Korean	105.5K	69.8K	9.1	1.4M
Russian-English	213.4K	68.7K	9.7	25.9M
English-Russian	76.2K	155.8K	9.5	25.9M
Spanish-English	107.1K	60.8K	9.8	61.4M
English-Spanish	73.9K	82.5K	9.7	61.4M
Thai-English	155.6K	84.2K	9.4	3.3M
English-Thai	109.2K	99.2K	9.2	3.3M
Vietnamese-English	96.6K	76.6K	9.0	3.5M
English-Vietnamese	96.4K	75.3K	9.3	3.5M

Table 2: Summary statistics for the *word2word* dataset between selected languages and English. Lexicon size refers to the number of unique words in source language for which translations exist. S.Chinese and T.Chinese refer to simplified and traditional Chinese, respectively.

335.5K (Arabic-English), demonstrating the broad coverage of words in the dataset. For each of these words, the dataset includes an average of 9 or more highest-scored translations according to our extraction approach described in Section 3.1. Lexicon size for all language pairs can be found in Appendix B.

### 2.2. Examples

In Table 3, we present samples of top-5 word translations in the English↔French and English↔Korean bilingual lexicons. For each language pair, we randomly sample five words from the top-10,000 frequent words in the source lexicon and provide their top-5 word translations. This is to show translations for words that are relatively more likely used than others in typical discourse.

### 3. Methodology

### 3.1. Bilingual Lexicon Extraction

Bilingual lexicon extraction (BLE) is a classical natural language task where the goal is to find word-level correspondences from a (parallel) corpus. There are many different approaches to BLE, such as word alignment methods (Brown et al., 1993; Vogel et al., 1996; Koehn et al., 2007) and cross-lingual word representations (Ruder et al., 2017; Mikolov et al., 2013a; Liu et al., 2013; Gouws et al., 2015; Conneau et al., 2017).

Among them, we focus on simple approaches that can work well with various sizes of parallel corpora that are present in OpenSubtitles2018, which ranges from 129 sentence pairs in Armenian-Indonesian to 61M sentence pairs in English-Spanish. In particular, we avoid methods that require high-resource parallel corpora (e.g., neural machine translation) or external corpora (e.g., unsupervised or semi-supervised cross-lingual word embeddings). Also, since bilingual word-to-word mappings are hardly one-to-one (Fung, 1998; Somers, 2001; Levy et al., 2017), we consider methods that yield relevance scores between every source-target word pair, such that we can extract not just one but the top-k correspondences. For these reasons, we consider approaches based on (monolingual and cross-lingual) co-occurrence counts: co-occurrences, pointwise mutual information (PMI), and co-occurrences with controlled predictive effects (CPE).

# 3.1.1. Co-occurrences

The simplest baseline for our goal is to compute the cooccurrences between each source word x and target word y. For each source word x, we can score any target word ybased on the conditional probability  $p(y|x) \propto p(x,y)$ :

$$p(y|x) = \frac{p(x,y)}{p(x)} \approx \frac{\#(x,y)}{\#(x)} \propto \#(x,y)$$
 (1)

where  $\#(\cdot)$  denotes the number of (co-)occurrence counts of the word or word pair across the parallel corpus. The top-k translations of source word x can be computed as the top-k target words with respect to their co-occurrence counts with x.

Word		Тор	-5 Translation	ıs	
English			French		
exceptional	exceptionnel	exceptionnelle	exceptionnels	exceptionnelles	exception
whether	plaise	décider	importe	question	savoir
committee	comité	éthique	accueil	commission	central
clown	clown	clowns	bouffon	guignol	cirque
spread	dispersez-vous	propagation	répandre	propager	répandu
French			English		
hobbs	hobbs	abigail	garret	jacob	garrett
mêlé	mixed	involved	middle	part	murder
établir	establish	establishing	set	able	connection
taule	slammer	joint	locked	jail	prison
chaussettes	socks	sock	stockings	pairs	underwear
English			Korean		
slaughtered	학살	도륙	도살	당했	 살육
shadow	그림자	그늘	알맞	어둠	존재
Charles	찰스	제프리	Charles	조프리	램퍼트
concerns	걱정	우려	염려	관한	판단력
reverse	역	뒤집	후진	거꾸로	되돌리
Korean			English		
아유	arm	Thank	thrilled	killing	NamWon
상어	shark	Shark	sharks	Tank	Tiger
쥐	rat	rats	mouse	mice	squeeze
기꺼이	willing	happy	pleasure	gladly	willingly
어떤	Some	kind	which	any	anything

Table 3: Randomly sampled words and their top-5 translations in the English↔French and English↔Korean *word2word* bilingual lexicons. Top-5 translations are listed in the descending order of scores.

# 3.1.2. Pointwise Mutual Information

Another simple baseline is pointwise mutual information (PMI), which further accounts for the monolingual frequency of a candidate target word y:

$$\mathsf{PMI}(x,y) = \log \frac{p(x,y)}{p(x)p(y)}$$

$$\approx \log \frac{\#(x,y)}{\#(x)\#(y)} \propto \log \#(x,y) - \log \#(y)$$
(2)

Compared to the co-occurrence model in (1), PMI can help prevent stop words from obtaining high scores.

The use of PMI has been connected to the skip-gram with negative sampling (SGNS) (Levy and Goldberg, 2014) model of *word2vec* (Mikolov et al., 2013b). PMI can also be interpreted as a conditional version of TF-IDF (Fung, 1998).

### 3.1.3. Controlled Predictive Effects

While conditional probability and PMI are proportional to cross-lingual co-occurrence counts, they can fail to distinguish exactly which source word in the sentence is the most predictive of the corresponding target word in the translated sentence. For example, given a English-French pair (*the apple juice*, *la jus de pomme*), these baseline methods cannot isolate the effect of *apple*, as opposed to *the* and *juice*, on *pomme*.

To deal with this issue, we add a correction term that averages the probability of seeing y given a confounder x'

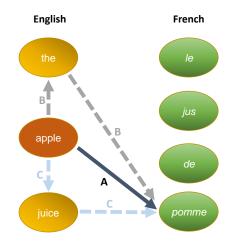


Figure 1: A schematic graphical model of English and French words. Co-occurrence and PMI models focus on the relationship from *apple* to *pomme* (A). CPE further controls for the confounding effect of other collocates like *the* (B) and *juice* (C).

in the source language, i.e. p(y|x'). This probability is then weighted by the probability of actually seeing that confounder, i.e. p(x'|x). This correction can be explained intuitively by the dashed arrows in the schematic graphical model in Figure 1– it reflects the conditional independence

Metric (%)	Method	en-es	es-en	en-fr	fr-en	en-de	de-en	en-ru	ru-en	en-zh	zh-en	en-it	it-en
# Sente	ence Pairs	61.	4M	41.	8M	22.	5M	25.	9M	4.8	3M	35.	2M
	Co-occurrence	22.3	25.5	18.7	21.9	10.5	23.5	3.3	11.4	5.4	3.8	24.9	24.1
D@1	PMI	72.7	72.3	73.9	72.1	62.1	71.9	32.8	55.0	24.8	33.1	68.1	69.5
P@1	MUSE	81.7	83.3	82.3	82.4	74.0	72.4	51.7	63.7	42.7	37.5	66.2	58.7
	CPE	82.4	79.5	83.6	80.7	82.4	81.1	66.7	68.9	56.0	<b>58.7</b>	80.9	82.1
	Co-occurrence	67.8	71.4	63.1	66.3	63.7	65.5	52.3	51.8	46.0	36.3	61.9	68.5
P@5	PMI	92.3	90.4	92.5	90.1	90.5	88.1	74.1	79.5	58.7	66.1	90.3	91.1
rws	MUSE	-	-	-	-	-	-	-	-	-	-	80.4	76.5
	CPE	90.1	88.4	91.7	89.3	90.7	87.7	79.5	80.0	73.5	72.8	89.8	89.9

Table 4: Precision (%) on 1,500 word translations (test split from MUSE) for language pairs evaluated in the MUSE paper. P@1 and P@5 denote the precision of top-1 and top-5 predictions, respectively. The ISO 639-1 language codes are used (en: English, es: Spanish, fr: French, de: German, ru: Russian, zh: traditional Chinese, it: Italian).

relationships between words that the baseline models do not. We call the resulting approach as the method of *controlled predictive effects (CPE)*.

Formally, we define the corrected CPE score as follows:

$$\begin{aligned} \mathsf{CPE}(y \mid x) &= p(y \mid x) - \sum_{x' \in \mathcal{X}} p(y \mid x') p(x' \mid x) \\ &= \sum_{x' \in \mathcal{X}} \mathsf{CPE}_{y \mid x}(x') p(x' \mid x) \end{aligned} \tag{3}$$

where  $\mathcal{X}$  is the source vocabulary and  $\mathsf{CPE}_{y|x}(x')$  denotes the CPE term of any other source word x' when predicting y from x. Formally, this term is defined as

$$\mathsf{CPE}_{y|x}(x') = p(y \mid x, x') - p(y \mid x') \tag{4}$$

This CPE term measures the effect of additionally seeing x (apple) when predicting y (pomme), after controlling for the effect of any other x' (the), which the model views as a confounder. If  $\mathsf{CPE}_{y|x}(x') = 0$ , then  $x \perp \!\!\! \perp y \mid x'$ , meaning that after observing a confounder x', x is no longer related to y. The CPE term for each confounder x' is then marginalized over all possible confounders to give a final score, weighted by the probability of seeing that confounder in a sentence with x. Note that  $\mathsf{CPE}_{y|x}(x) = 0$ , meaning that, after seeing x when predicting y, there is no additional effect by seeing x (again).

In practice, summing the CPE scores over all words in the source vocabulary can be inefficient. Because many of the (unrelated) words in the vocabulary do not play a role in the confounding, we select the top-m source words with the highest co-occurrence counts and correct for their effects only. We used m=5,000 in our experiments and found that using a larger m did not make a meaningful difference on the quality of top-1 and top-5 correspondences.

### 3.1.4. Evaluation on MUSE Bilingual Dictionaries

We first evaluate the methods on the same ground-truth bilingual dictionaries as MUSE<sup>4</sup>, a cross-lingual neural embedding model. Each dictionary contains 1,500 words and their translations obtained using an internal translation tool from the authors. Although we consider MUSE's performance as a reference, we do note that it is difficult to make

a fair comparison against MUSE: the count-based methods use parallel corpora from OpenSubtitles2018, while MUSE embeddings are instead learned from monolingual Wikipedia data (for its unsupervised version) and an additional 5,000-word bilingual lexicon (for its supervised version).

In Table 4, we report the top-1 and top-5 precision scores (P@1 and P@5, respectively) of the count-based methods and MUSE embeddings across twelve<sup>5</sup> directed language pairs that were used to evaluate MUSE in its paper (Conneau et al., 2017): English-Spanish, English-French, German-English, English-Russian, English-Chinese (traditional), and English-Italian, all in both directions. For MUSE, we report its best reported performance (only top-1 precision is reported, except for en-it and it-en) among its supervised and unsupervised variants.

Our main finding is that the CPE method consistently and significantly outperforms the co-occurrence and PMI baselines at top-1 precision score. We also find that CPE outperforms MUSE on most of the reported language pairs, especially when the number of sentence pairs is comparatively small (e.g., 13-21% improvement between English and Chinese, for which there are about 6% as many sentence pairs as those between English and Spanish). In terms of the top-5 precision score, the CPE method performs comparatively well with the PMI method, which performs better on some of the selected language pairs. Compared to the CPE method, we suspect that the PMI method overly favors rare words because it directly penalizes word counts, so that the most likely correspondence (which isn't necessarily the least common) is pushed back to later ranks. More examples can be found in Appendix A

#### 3.1.5. Evaluation on Non-European Languages

Next, we compare the performance of co-occurrence, PMI, and CPE methods on language pairs between English and some of the major non-European languages: Arabic, simplified Chinese, Japanese, Korean, Thai, and Vietnamese. As we detail in Section 3.2., these languages commonly require special word segmentation techniques. Also, they

<sup>4</sup>https://github.com/facebookresearch/MUSE

<sup>&</sup>lt;sup>5</sup>The MUSE paper also presents the results on English-Esperanto and Esperanto-English, but the ground-truth dictionary is no longer available online. See https://github.com/facebookresearch/MUSE/issues/34.

Metric (%)	Method	en-ar	ar-en	en-zh	zh-en	en-ja	ja-en	en-ko	ko-en	en-th	th-en	en-vi	vi-en
# Sente	ence Pairs	29.	8M	11.	2M	2.1	M	1.4	ŀМ	3.3	3M	3.5	5M
	Co-occurrence	23.3	1.1	2.1	0.4	5.0	0.3	22.9	0.4	0.6	0.5	4.0	2.1
P@1	PMI	13.3	20.7	8.5	20.6	33.5	16.7	14.0	14.9	18.3	13.4	20.5	16.5
	CPE	30.3	27.9	48.3	34.3	49.3	40.4	39.1	38.1	48.1	31.0	30.0	37.7
	Co-occurrence	46.9	35.2	50.5	27.1	30.7	29.1	36.6	26.9	55.6	24.4	39.3	28.3
P@5	PMI	57.0	61.6	78.7	65.3	64.0	60.5	48.8	57.7	64.5	52.7	50.1	60.4
	СРЕ	58.1	50.5	80.9	60.1	66.8	66.4	54.9	60.0	69.3	53.1	48.9	62.2

Table 5: Precision (%) on 2,000 word translations between six *non-European* languages and English (source words randomly sampled from OpenSubtitles2018; gold labels taken from Google Translate). P@1 and P@5 denote the precision of top-1 and top-5 predictions, respectively. The ISO 639-1 language codes are used (ar: Arabic, zh: simplified Chinese, ja: Japanese, ko: Korean, th: Thai, vi: Vietnamese).

Language	Python Tokenizer Module	Reference
Arabic	pyarabic.araby	(Zerrouki, 2010)
Chinese (Simplified)	Mykytea	(Neubig et al., 2011)
Chinese (Traditional)	jieba	n/a
Japanese	Mykytea	(Neubig et al., 2011)
Korean	konlpy.tag.Mecab	(Park and Cho, 2014)
Thai	pythainlp	n/a
Vietnamese	pyvi	n/a
Others	nltk.tokenize.TokTokTokenizer	(Bird et al., 2009; Dehdari, 2014)

Table 6: List of Python tokenizer modules used for each language.

typically have relatively smaller amounts of sentences paired with English, making it more challenging for the models to achieve high precision.

Unfortunately, we learned in our early experiments that the MUSE test set translations are far from being perfect for these non-European languages. For example, in English-Vietnamese, we found that 80% of the 1,500 word pairs in the test set had the same word twice as a pair (e.g. crimsoncrimson, Suzuki-Suzuki, Randall-Randall). Thus, for the non-European languages, we instead evaluate on translations using Google Translate<sup>6</sup>, a proprietary<sup>7</sup> web software for machine translation. To construct this test set, we first sample 2,000 words from the monolingual word distribution of that language pair's OpenSubtitles2018 parallel corpus. We use temperature-based smoothing (T = 1.25) for the distribution to include more low-frequency words in the test set and also filter out words that include characters not from its alphabet (e.g., Charles in Korean). Then, for each of the 2,000 sampled words, we retrieve "common" and "uncommon" translations<sup>8</sup> from Google Translate and treat them as ground truth labels.

The results are summarized in Table 5. Here, we see more evidence that the CPE method performs significantly better than both the co-occurrence and the PMI methods in top-

1 precision as well as top-5 precision. The performance gap tends to be larger both when the language's words are not whitespace-separated (e.g., Chinese and Japanese) and when there are a relatively small number of paired sentences (e.g., Korean and Thai). Based on the results from Tables 4 and 5, we employ the CPE method to produce the *word2word* dataset.

## 3.2. Word Segmentation

Since many of the 62 languages we consider are sensitive to word segmentation, we use language-specific tokenization tools when necessary. Specifically, we use publicly available tokenization packages for morphologically complex languages, i.e., Arabic (Attia, 2007) and Korean, and languages in which words are not separated by spaces, i.e., Chinese, Japanese, Thai, and Vietnamese<sup>9</sup>. For all other languages, we use the tok-tok tokenizer (Dehdari, 2014) implemented in NLTK (Bird et al., 2009). Table 6 summarizes the tokenization packages we used in the *word2word* dataset and their references.

### 4. The word2word Python Interface

As part of releasing the dataset and making it easily accessible and reproducible, we also introduce the *word2word* Python package. The open-source package provides an easy-to-use interface for both downloading and accessing bilingual lexicons for any of the 3,564 language pairs and building a custom bilingual lexicon on other language pairs for which there is a parallel corpus. Our source code is available on PyPi as https://pypi.org/project/word2word/.

https://translate.google.com/

<sup>&</sup>lt;sup>7</sup>We note that, because Google Translate is proprietary and not open-source, its results may change depending on the time of access. Our evaluations use Google Translate results accessed on July 19, 2019.

<sup>&</sup>lt;sup>8</sup>For word translations, Google Translate categorizes its translations to three categories: common, uncommon, and rare translations.

<sup>&</sup>lt;sup>9</sup>Spaces in Vietnamese delimit syllables.

# 4.1. Implementation

The word2word package is built entirely using Python 3. The package includes scripts for downloading and preprocessing parallel corpora from OpenSubtitles2018, including word segmentation, and for computing the CPE scores for all available word tokens within each parallel corpus. After processing, the package stores the bilingual lexicon as a Python pickle file, typically sized a few megabytes per language pair. The pickle file contains a Python dictionary that maps each source word to a list of top-10 word correspondences in O(1) time. This allows bilingual lexicons to be portable and accessible.

## 4.2. Usage

The Python interface provides a simple API to download and access the *word2word* dataset. As demonstrated in Figure 2, word translations for any query word can be retrieved as a list with a few lines of Python code.

```
from word2word import Word2word
en2fr = Word2word('en', 'fr')
print(en2fr('apple'))
# ['pomme', 'pommes', 'pommier',
# 'tartes', 'fleurs']
```

Figure 2: The *word2word* Python interface for retrieving word translations.

### 4.3. Building a Custom Bilingual Lexicon

The word2word package also allows training a custom bilingual lexicon using a different parallel corpus. This can be useful in cases where there are larger and/or higher-quality parallel corpora available for the language pair of interest or when utilizing word translations for a particular domain (e.g., government, law, and medical). This process can also be done using a few lines of Python code, as demonstrated in Figure 3. For an OpenSubtitles2018 corpus of a million parallel sentences, building a bilingual lexicon takes approximately 10 minutes using 8 CPUs.

```
from word2word import Word2word

my_en2fr = Word2word.make(
    'en', 'fr', 'data/pubmed.en-fr'
)
# ...building...done!
print(my_en2fr('mitochondrial'))
# ['mitochondriale', 'mitochondriales',
# 'mitochondrial', 'cytopathies',
# 'mitochondriaux']
```

Figure 3: The *word2word* Python interface for building a custom bilingual lexicon. Once built, the lexicon can be accessed in the same way as done in Figure 2.

## 5. Conclusion

In this paper, we present the *word2word* dataset, a publicly available collection of bilingual lexicons for 3,564

language pairs that are extracted from OpenSubtitles2018. The bilingual lexicons have high coverage (up to hundreds of thousands words) for many language pairs and provide word translations of similar or better quality compared to those from a state-of-the-art embedding model. We also release the *word2word* Python package, with which the user can easily access the dataset or build a custom lexicon for different parallel corpora. We hope that the dataset and its Python interface can facilitate research on improving cross-lingual models, including machine translation models (Ramesh and Sankaranarayanan, 2018; Gū et al., 2019) and cross-lingual word embeddings (Conneau et al., 2017; Ruder et al., 2017).

# 6. Bibliographical References

Artetxe, M., Labaka, G., and Agirre, E. (2018). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia, July. Association for Computational Linguistics.

Artetxe, M., Labaka, G., and Agirre, E. (2019). Bilingual lexicon induction through unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5002– 5007, Florence, Italy, July. Association for Computational Linguistics.

Attia, M. A. (2007). Arabic tokenization system. In *Proceedings of the 2007 workshop on computational approaches to semitic languages: Common issues and resources*, pages 65–72. Association for Computational Linguistics.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language* processing with Python: analyzing text with the natural language toolkit. "O'Reilly Media, Inc.".

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jégou, H. (2017). Word translation without parallel data. *arXiv* preprint arXiv:1710.04087.

Dehdari, J. (2014). *A Neurophysiologically-Inspired Statistical Language Model*. Ph.D. thesis, The Ohio State University.

Fung, P. (1998). A statistical view on bilingual lexicon extraction: from parallel corpora to non-parallel corpora. *Machine Translation and the Information Soup*, pages 1–17.

Glavaš, G., Litschko, R., Ruder, S., and Vulić, I. (2019). How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 710–721, Florence, Italy, July. Association for Computational Linguistics.

Gouws, S., Bengio, Y., and Corrado, G. (2015). Bilbowa: Fast bilingual distributed representations without word alignments. In Francis Bach et al., editors, *Proceedings* 

- of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research, pages 748–756, Lille, France, 07–09 Jul. PMLR.
- Gū, J., Shavarani, H. S., and Sarkar, A. (2019). Pointer-based fusion of bilingual lexicons into neural machine translation. *arXiv preprint arXiv:1909.07907*.
- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.
- Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185.
- Levy, O., Søgaard, A., and Goldberg, Y. (2017). A strong baseline for learning cross-lingual word embeddings from sentence alignments. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 765–774.
- Lison, P., Tiedemann, J., and Kouylekov, M. (2018). Opensubtitles 2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Liu, X., Duh, K., and Matsumoto, Y. (2013). Topic models + word alignment = a flexible framework for extracting bilingual dictionary from comparable corpus. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 212–221, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Mikolov, T., Le, Q. V., and Sutskever, I. (2013a). Exploiting similarities among languages for machine translation. *arXiv* preprint arXiv:1309.4168.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Neubig, G., Nakata, Y., and Mori, S. (2011). Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 529–533. Association for Computational Linguistics.
- Park, E. L. and Cho, S. (2014). Konlpy: Korean natural language processing in python. In *Proceedings of the* 26th Annual Conference on Human & Cognitive Language Technology, Chuncheon, Korea, October.
- Ramesh, S. H. and Sankaranarayanan, K. P. (2018). Neural machine translation for low resource languages using bilingual lexicon induced from comparable corpora. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 112–119, New Orleans, Louisiana, USA, June. Association for

- Computational Linguistics.
- Ruder, S., Vulić, I., and Søgaard, A. (2017). A survey of cross-lingual embedding models. *arXiv preprint arXiv:1706.04902*.
- Somers, H. (2001). Bilingual parallel corpora and language engineering. In *Proc. Anglo-Indian Workshop*" *Language Engineering for South-Asian languages*.
- Vogel, S., Ney, H., and Tillmann, C. (1996). Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 836–841. Association for Computational Linguistics.
- Vulić, I. and Moens, M.-F. (2013). Cross-lingual semantic similarity of words as the similarity of their semantic word responses. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 106–116, Atlanta, Georgia, June. Association for Computational Linguistics.
- Zerrouki, T. (2010). pyarabic, an arabic language library for python.

# A Sample Translations from Different Extraction Methods

In Table 7, we compare the BLE methods described in Section 3.1. from illustrative examples of their extracted bilingual lexicons for English to Spanish and English to Simplified Chinese. These examples show that the CPE approach provides the correct correspondence as its top-1 translation in both languages, while the PMI approach seems to excessively favor rarer words among the co-occurrences. As illustrated in the English-Chinese example, this can be particularly problematic with languages such as Chinese, where word segmentation is highly nontrivial. The co-occurrence method prefers stop words that are frequent over the entire document, rather than the corresponding words.

#### A1. Co-occurrences

The baseline co-occurrence model performs poorly in both experiments (Tables 4 and 5). As exemplified in Table 7, we find that the top-5 predictions in many cases are primarily stop words, such as la (the), de (of), and que (that) in Spanish and 的 (of), 你 (you), and 我 (I, me) in Chinese, because they frequently occur in any sentence, regardless of context.

# A2. Comparing PMI and CPE

Comparing translations using PMI and CPE, we find in Table 7 that PMI favors less frequent words excessively. This results in two kinds of error cases: (a) when PMI overemphasizes rare words in the target vocabulary, e.g. *solarización* for *library* in en-es, and (b) when PMI misses correct words in the target language that are relatively frequently used, e.g. *bien* for *good* in en-es. Another consequence is that PMI prefers less common variants of the same word, in particular conjugations and past/future tenses as well as typos, when two forms of the same word have comparable counts (e.g. *obligados* preferred over *obligado* in Spanish for the English *obliged*).

Because of the second reason, we also find that *word2word* tends to be more robust to tokenization issues, which are common in non-whitespace-separated languages like Chinese. For example, since the tokenizer failed to separate 张 开嘴 (open mouth), which in general occurs far less frequently than 嘴 (mouth), PMI favors 张开嘴 over the more frequent 嘴 as its first choice.

## **B** Full Dataset Statistics

In Table 8, we list the sizes of all 3,564 bilingual lexicons in the *word2word* dataset. By size, we refer to the number of source words for which translations exist. For each source word, we extract up to 10 (9+ on average) most likely translations according to the CPE method described in 3.1.3.

English	Methods	To	op-5 Transl	ations in S	panish		Top-5 Tr	anslatio	ns in Sim	plified (	Chinese
	Co-occurrence	de	la	que	el	у	的	它	了	- 是	我
its	PMI	propia	sus	su	tierra	poder	它	政府	国家	失去	由
	CPE	su	sus	propia	tierra	cada	它	将	自己	国家	中
	Co-occurrence	que	de	no	bien	es	好	的	你	我	很
good	PMI	buenas	noches	buenos	buena	buen	祝你好运	晚安	好消息	早上好	早安
	CPE	bien	buena	buenas	buen	bueno	好	很	不错	晚安	早上好
	Co-occurrence	la	boca	de	que	no	的	你	我	嘴	了
mouth	PMI	boca	cerrada	pico	mantén	abre	张开嘴	嘴里	大嘴巴	张嘴	嘴巴
	CPE	boca	cerrada	abre	palabras	labios	嘴	嘴里	嘴巴	闭上	闭嘴
	Co-occurrence	la	biblioteca	de	en	que	图书馆	的	我	在	你
library	PMI	solarización	biblioteca	soltándola	library	librería	英	图书馆	圖書館	藏书室	书房
	CPE	biblioteca	la	librería	pública	tarjetas	图书馆	书房	里	圖書館	去

Table 7: Selected *word2word* translations of English words into Spanish and simplified Chinese. Top-5 predictions are listed in the decreasing order of the model's scores. Boldfaced target words indicate correct translations.

3K 3Z6K 223K	63K	0 283K	33K	166K	174K	106K	9K	153K	110K	216K	343K	131K	22K	16K	261K	3K	134K	131K	159K	63K	0	49K	213K	199K	132K	91K	140K	269K	162K 140K	183K	325K	274K	228K	319K	178K	17K	9K	2K	305K	114K	104K	43K	176K	0
6K 3K 6X 3K 334K 326 7220K 223I	78K	6K K 278K	51K	X 171K	167K	K 102K	15K	X 149K	100K	208K	X 342K	126K	24K	19K	K 260K	4K		177K	K 153K	77K	2K	28K	269K	226K	185K	114K	139K	X 270K	X 160K	X 185K	309K	289K	X 232K	309K	K 181K	19K	12K	8 8 8		169K	107K	X 67K	0	X 191K
en ze_zh K 0 IK 302K 4K 188K	K 10K	¥	K 21K		105K 92K		K 4K	IK 153F	701 X	K 76K	5K 273F	V	7 4K	K 805	199K 229K	0 25 N	~			18K 22K	4 0	14K 16K	K 43K	41K 57K	-	K 24K	78K 78K	JK 278F	9K 142F	3K 179F	37K 171K	-		220K 263K	2K 122F	-	1K 0 35K 43K	_		30K 28K	8	116K	7K 201F	63K 197F
vi ze. 3K 11 09K 23 06K 13	69K 11	0 0 74K 128	6K 22	51K 115K	53K 10:	96K 11	1 1	41K 14	1K 14	10K 52	27K 25	20K 101B	18K	4K	45K 19	0 0	×		50K 11	3K 18k	2K 0	9K 14	30K 31	95K 41	18K	4K 28	131K 12	53K 26	48K 20K 12	71K 16	90K 13	58K 98	15K 15;	19K 58K	56K 12	1K 3K	3K 13	7K	294K 279K	19K 30	0 37	45K 0	17K 10	98K 63
0 3 26K 30 20K 20	6K 6	0 13K Z	0 4 23K 24	12K 18	14K 10	14K 9	0	17K 1	317 8	17K 2	22K 33	14K I	2K 2	2K 1	22K 2	% NC7	13K	2K	14K 15	6K 6	4 0	6K	7K 15	13K 19	2K	10K	13K I		17K 14	18K I	21K 2		19K 2	26K 29		0	0 X	0	27K 29	3K	7.	0 0	17K 2	13K 19
uk 7 3K K 241K K 161K	X 21K	X 2K K 122K	K 23K	X 86K	K 103K	74K	5 SK	K 125K	N 13/K	K 85K	K 225K	X 94K	8 8 X	4 4 K	K 192K	V 7 C	٠,	K 28K	K 110K	7 26K	2K	Z 20K	K 63K	K 69K	K 21K	K 28K	X 100K	K 212K	K 113K	K 145K	K 214K	K 101K	K 146K	K 237K	K 95K	ш	0 X	2K	2	0 25	36K	K 12K	K 105K	K 59K
11 tr 0 13F 12K 33S 10K 193	3K 87F	0 10F 5K 259	0 90F	3K 155	11K 145	10K 86K	3K 28K	2K 134	10 1 X	K 219	10K 327	11K 110	1K 241K	790 22F	11K 235	0 8 X		K 243	2K 138	6K 78F	0 4K	1K 83F	30 I	K 217	7K 246	K 115	7K 122 3K 166	2K 245	0K 146	1K 165	15K 285	2K 279	OK 222	1K 270	6K 164	$\vdash$	0 13K	0 12F		3K 291K	6K 97F	0 1111	K 216	2K 202
2K 2 327K 1	48K	0 280K	40K	151K 3	171K 1		8K	151K 1	47K	194K	334K 1		19K	13K 7		3458	×			37V (	2K		136K	162K	77K	89K	131K	262K 1	145K 1	167K 1	285K 1		218K 1	310K	166K		2K			98K		31K	223K	151K 2
7 te	0	0 K 13K	0 K	12K 12K	X X	14K 12K	0	K 16K	13K 14K	X 9K	K 18K	9K	U N	К 9К	20K	4 0		0 ×	K 9K	7K 4K	0	X IK	X 3K	38	0		X X	K 18K	х х 7 7 7 7	19K 17K	K 13K	11K 10K	X 14K	2 K	K 10K		0 2 2	0	20K	0 0	7K 6K	X 982	K 13K	K 7K
sv ta 6K 1K 51K 26K 115K 18K	32K 31	7K 0 80K 13F	9K 0	60K 12	59K 12	98K 14	17K 0	44K 18	CI NIO	23K 12	38K 21K	108K 120K 12K	63K 2	20K 12	55K 21K	317 AC18			46K 10	76K 71	5K 0	71K 6K	16K IS	28K 7K	99K 6K		132K 12K 173K 8K	66K 20	61K 14 40K 15	85K 19	06K 13K	84K 11	28K 15	15K 11K 06K 24K	0		11K 12K	7K 0	315K 22K	214K 0	05K 71	35K 2K	19K 15	05K 15
344K 3	83K 8	9K 254K 2	83K (	157K 1	148K 1	86K	26K	130K 1	2 N + C 2	220K 2	329K 3		247K 2	22K	233K 255K	345K	129K	246K 2	138K 1	78K	SK SK	81K	30/K 3	216K 2	227K 1	115K 1	120K	248K	149K 1	169K 1	286K 3	277K 2	221K 2	210K			13K	¥	291K	275K 2	101K	87K 9	218K 2	203K 2
sq 1K 330K 196K	-	5K 261K	24K		X 166K	4 98K	¥	X 140K	7.7K	X 187K	K 323K	121K	12K	12K		425 425		K 121K		41K		(1)	85K				K 127K	K 256K	K 143K	X 157K	288K		212K	295K	K 165K		9K			78K	76K	23K	X 216K	K 133K
s sk K 8K 8K 340H 8K 207H			K 74K		71K 158K	K 102K	ζ 20K	K 147F	15C2 A	K 220F	5K 335K		19K 3/K	K 18K		7 585	IK 130	142K 243F		K 77K	_	K 73K	K 308	K 221K			134K 132K	5K 260F	156K 154K 137K 137K	7K 180K	7K 292K		3K 0	288K 298K	175K 168K	$\vdash$	8K 12K			3K 239K	100K 103F	K 86K	K 2081	6K 198F
si sk 11K 5K 196K 348K 125K 209K	V	0 5K 108K 273K	8K 41K	در الصدر ال	80K 171	73K 106K			35K 90K		172K 336K		0K 19	8K 14K		200K 357K		30K 142		28K 62K	_		70K 214	74K 206K			77K 134K		92K 156 85K 137	112K 177K	148K 297K		64 .	200K 288K	89K 175	$\vdash$	3K 8K	1		25K 133K		8K 43K	95K 216	69K 196
10K 332K 19 207K 12	77K 2	8K 274K 10	74K 8		164K 8	96K	24K	141K 9	202K 1.	210K 9	344K 17	124K 7	41K	17K	248K 15	3K		212K 3	151K 7	77K 2		79K 2	293K 3	214K 7	180K	107K	139K 7	261K 15	158K 9	186K 1	0 12	289K 7	222K 11	293K 20	177K 8		9K	_	306K 19	263K 2	103K 3	8 NO8	218K 9	207K 6
16K 343K 194K	88K	10K	387K	158K	214K	89K	27K	132K	N/C7	220K	328K	109K	244K	24K	236K	8K	130K	244K	140K	78K	4K	24K	303K	219K	242K	117K	117K	247K	148K	0	165K	273K	221K	271K	163K	20K	14K	9K	290K	1297K	7 36K	118K	217K	200K
pt.br 7 16K K 347K K 196K	X 90K	K 262K	K 95K	K 161K	K 148K	K 99K	7 31K	K 136K	X 108K	K 222K	K 331K	K 112K	K 246K	Z 25K	K 230K	AN SOUR		K 247K	K 140K	79K	4K	X 85K	X 307K	K 225K	K 254K	X 117K	K 118K	K 249K	147K	172K	K 270K	K 280K	K 227K	279K 261K	K 163K	X 20K	K 13K	Z 11K	.K 295K	7 306k	K 100K	K 123K	K 216K	K 205K
pl pt 10K 12F 353K 346 209K 199	87K 84K	10K 10K	9K 85	161K 159	155K 157	93K 89F	6K 221	37K 136	3C7 NOC	1K 216	12K 334	16K 115K	7K 245	23K 18F	245K 232K	7K 6K		0K 245	148K 147	9K 76	K SK	3K 83	OK 169	0K 222	4K 218		129K 125K 176K 169K		157K 0 130K 129K	172K 168K	3K 291	9K 279	30K 221	2K 279	167K 167	16K 15F	3K 13	K 10	298K 294K	J5K 266	3K 101	32K 107	4K 215	16K 202
5K 10 355K 35 216K 20	-	8K 10	57K 89		161K 15	95K 93	12K 2	144K 13	07 NC07	114K 23	31K 33	120K 117K 121K 116K	5 XC2	16K 23	249K 24	3/4K 30	0 13	22 X8Z		76K 79	4K 6	8. 8.	270K 31	16K 23	146K 22	101K 11	132K 0 17	268K C	37K 13	178K 17	07K 28	386K 28	228K 25	308K 288K 300K 282K	171K 16		10K 13	4K 8	304K 29	101K 28	03K 10	58K 15	27K 22	202K 20
12K 364K 3	82K		86K	162K	156K	91K	19K	142K	7602 176K	224K 2	339K 3	117K	41K	21K	242K	3K	117K 134K	252K	149K	78K			314K 2		202K	117K	0 173K	264K	142K 156K 1 125K 132K 1	166K 171K 1	270K 308K 3	188K 288K 2	232K 2	288K	172K		13K		306K	277K 1	103K	115K	225K	204K 2
ms 1K X 321K X 194K	_		22K		7 157K	, 93K	4K		238K	207K	X 313K	120K	12K 7 215K	13K		0 32/K				25K	_	_	165K	_		_					-			308K	159K	$\vdash$	5K	_		76K	7 90K	27K	217K	159K
K 3K 5K 159K 3K 98K	.К 30K	1K 2K 272K 85K	42K 7K		168K 69K	104K 59K	K 2K	5K 79K	/N 88K	7K 67K	7K 126K	122K 62K	K 0K	X A	246K 109K	520 154	-	9K 24F	5K 64K	K 18K	0	14K	5K 55 K	62K	120K 0		133K 56K 171K 50K		153K 75K	179K 92K	297K 111K		3K 90F	1K 148K	3K 67K	$\vdash$	3K 0		296K 158K	4K 18K	95K 29F	7K 7K	5K 76F	4K 60K
1v mk 1K 4K 97K 316F 33K 213F	23K 57	0 1 122K 27	10K 42		94K 16	67K 10	2K 2	98K 14	42 A C 4	36K 20	97K 32	78K 12	42 K 24	Ή Ξ	154K 24	C6 A61	×	50K 13	39K 15	25K 58	2K +5	20K 4(	24K 20	73K	4K 12		82K 13 81K 17		88K 13	13K 17	166K 29		30K 22	04K 10	39K 17	$\vdash$	4K 5K 3	0 5	192K 29	36K 11	40K 99	9K 23	96K 22	78K 17
972 328K 1 206K 1	41K	0 215K 1	23K	153K	161K	98K	3К	141K	1 NIC2	156K	323K 1	123K	219K	12K	243K 1	343K	113K	111K	146K	39K	2K	31K	0 108K	141K	73K	63K	130K	260K	147K 1	164K 1	274K 1	186K	214K 1	308K	167K	12K	9K	2K	287K 1	67K	4R 4	14K	195K	137K
0 0 5 336K 5 211K	36K	0 K 155K	21K	X 126K	X 145K	K 106K	ζ 2K	K 145K	A3K	K 152K	307K	K 122K	711K	IIK	X 272K	0 349K	88 K		K 135K	70K	0	0	43K	85K	42K	49K	K 133K	K 264K	K 147K	X 169K	38K	701X	X 186K	40K	K 144K	11K	1K	2K	X 297K	51K	O.R.	15K	701X	108K
ka kk 0 0 102K 2K 73K 4K	8K 0	0 0 65K 2K	6K 0	44K 3K	45K 4K	41K 3K	0	57K 3F	01K 2	45K 2F	94K 5F	43K 2K	3K U	1K		0 4K	~		42K 3F	15K 4F	0	11K 0	31K 2F	42K 0	12K	15K 11	37K 3K		54K 4F 50K 3F	65K 3F	74K 4F	33K 2F	68K 3F	30K 0		0	0 0 24K	2K 0		15K 2K	19K 11	2K 0	42K II	26K 0
ja 2K 336K 205K	52K	0 241K	40K		166K	102K	Ж.		NUCZ NUCZ		334K	124K	18K	15K		AUCE O	×	61K	138K	0 312	2K	45K	140K	144K	75K	70K	133K		146K	¥	289K	192K	223K	344K	167K		6K		299K	103K	81K	25K	230K	152K
it 10K K 346K K 205K	X 77K	8K K 276K	X 86K	K 163K	K 157K K 228K	X 95K	23K	K 138K	X 230A	X 212K	K 328K	K 116K	X 252 K	19K	K 245K	N 304 N	× 130K	247K	K 0	X 79K	4 ¥	X 84K	K 303K	K 213K	X 185K	X 111K	K 131K K 177K	K 268K	K 158K	K 175K	K 291K		K 224K	K 272K	K 169K	11K	9K	Ξ	K 299K	X 270K	X 102K	X 91K	K 218K	K 202K
id is 4K 0 319K 331 211K 200	85K 231	7K 5K 265K 216	60K 181	157K 158	237K 220	1K 981	13K 2K	- 0	N647 NIC7	3K 991	326K 329		25K 0K	17K 4K	3K 246	455 0	0 92K	175K 0	****	78K 24K	2K 2K	1K 18	9K 148	0K 128	2K 411	2K 501	4K 132	(1)	4K 151 3K 131	6K 170	3K 249K		0K 216	6K 108	1K 171	3K 3K	10K 0	7K	7K 294K	3K 371	102K 521	366K 151	213K 153	192K 106
0 K	0	0 00	0 1	10	<del>\$</del> <del>\$</del>	4K	2K 1:	6K 14	488 23	0	3K 32	2K	٥ پز	0	246K 5K 243K	A 0	394		787 15	0 0	0 0	0	0 0	51421	129	0	0 K	7K 25	4 K	7K 17	3K 29	195 26	541 22	399 IS	2K 17	0	0 0	0	8K 29	0 0	0 0	0 0	3K 21	3K 19
gl he hi hr hu hy 0 9K 1K 9K 11K 0 70K 347K 58K 343K 350K 6K 45K 213K 36K 199K 202K 2K	X 86K	K 271K	5K 82K 88K 44K 260K 258K	23K 166K 24K 159K 159K	36K 162K 28K 151K 156K 66K 251K 38K 214K 222K	X 94K	K 24K 2K	133K 136K	N 727 N	K 217K	51K 340K 45K 331K 332K	K 116K	248K 248K	21K 21K	246K	> ¥	K 131K	K 243K	K 146K	77K	4 K	X 82K	K 303K K 167K	K 218K	K 222K	K 115K	K 128K	K 258K	K 157K	K 169K	K 283K	K 276K	K 226K	K 269K	K 168K	X 16K	0 13K 12K 13K 12K 0	10K	K 293K	K 285K	100K 100K	X 103K	K 218K	K 201K
hi hr 1K 9K 58K 343K 36K 199K	10K 85K	0 9K 26K 261K	5K 82K	4K 159	8K 151	6K 891	2K 20K	33K 133	0C7 N7	7K 216	5K 331	6K 110	40K 248K	0 211	5K 0	71K 308K 30K 300K	0K 129	5K 246	6K 142	12K 77K	0 4K	0K 831	3K 306	6K 217	9K 210	4K 114	4K 123	4K 259	7K 148 0K 124	6K 166	5K 287	7K 275	0K 222	OK 210	6K 165	9K 16I	2K 131	346 8K	0K 292	3K 267K	3K 100	2K 921	4K 217	5K 203
9K 347K 5 213K 3	85K 1	9K 272K 2	83K	166K 2	36K 162K 28K 66K 251K 38K	104K	19K 2K	74K 150K 3	1677	218K 2	340K	127K 2	34K	21K	253K 4	3K	132K 2	248K	148K 2	76K 1	4K	84K 1	307K 2	222K	210K	113K 1	135K 180K 2	266K 4	156K 2 143K 3	185K 3	290K 3	285K 2	227K 3	319K S	176K 2	20K 1	13K 1	8K	308K 5	263K	102K 1	70K	210K 3	201K 2
fr         gl         he         hi           11K         0         9K         1K           347K         70K         347K         58K           205K         45K         213K         36K	K 8K	1K 38K	12K	1K 23K	2K 36K	K 51K	K 544	4K 74K	K 14K	1K 25K	NK 51K	4 A	O 40 K	K 987	5K 57K	Y 0	1K 17K	1K 5K	2K 49K	7 10K	2K 0	K 10K	X X	K 22K	3K 9K	4K 9K	3K 28K K 19K	JK 70K	4K 43K	1K 48K	5K 56K	)K 24K	9K 42K	X 79 K	5K 24K	0	K 0	0	1K 98K	8K 12K	K 12K	1K 3K	K 32K	)K 24K
6 fr 8K 11 353K 347 211K 209	82K 80K	9K 10K 276K 267K	7K 86	50K 16	55K 152	3K 91	8K 23K	40K 132	20 NSC 72K 18	21K 22	0 339K	17K 0	7K 05	1K 21	51K 24:	3K 30	32K 13-	52K 249	44K 142	77K 79K	5K 21	9K 83	13K 31	26K 22	02K 22	15K 112	30K 12:	53K 26(	57K 15/ 35K 129	79K 17	38K 280	83K 29(	29K 229	18K 26	70K 16	9K 18	3K 10	7K 11	11K 30	34K 268	24K 10	5K 10	16K 22	04K 209
fa         fi           4K         8K           307K         353K           220K         211K	72K 8	5K 9 275K 27	49K ;	155K 1	172K 1	107K 9	8K 18K	149K 130K 151K 140K 134K	143N 235N 235N 202N 36N 25/N 32N 230N 23/N 3	0 2	340K	127K 1	235K 2	18K 2	269K 2	300K 3	134K 13	116K 2	156K 1	72K .	2K 5	60K	104K 16	216K 2	148K 2	110K	136K 1	267K 2	155K 1	180K 1	300K 3	265K 28	227K 2	320K 2	170K	12K	8K	8K	304K 3	148K 2	105K 10	37K 5	211K 2	189K 2
eu 0 236K 162K	75K 24K	1K 123K	19K	92K	112K	89K	4K	130K	145K	95K	236K	99K	15K	Ж	179K	N +/ 7	63K	35K	110K	30K 72K	0	22K	34K	63K	51K	30K	102K 86K	206K	123K	150K	186K	105K	148K	231K	113K	SK	34K	3K	252K	42K	38K	11K	115K	96K
cs ct cu fra fi fr gi he hi hr hu 16K SK	K 75K	K 266K	96K 66K 19K 49K 77K 86K	K 161K	K 162k	K 103K	K 18K	149K	N 120K	K 210K	K 330K	K 124K	K 230K	K 18K	K 250K	400c V	K 128K	K 242K	K 149K	K 75K	ZK OZR	K 60K	K 306k	K 218K	K 162K	K 110k	X 134 X	K 257K	K 132k	K 180K	K 297h	K 275K	K 223K	K 306K	K 172K	K 12K	14K 11K 1K 8K 13K 10K	X 3K	K 296K	K 200k	K 105K	K 68K	K 208K	K 198K
		5K         10K         8K         9K         10K         10K         2K         10K         9K         1K         5K         9K         10K           93K         260K         259K         259K         253K         13K         262K         266K         123K         276K         276K         57K	3K 250	160K 157K 151K 9K 159K 161K 92K 155K 160K 162K	143K 142K 18K 147K 162K 112K 172K 155K 152K 0 197K 30K 211K 231K 178K 246K 255K 218K	0 22K 97K 103K 89K 107K 93K 91K 51K 104K 26K 89K 94K 4K 101K	0 34K	30K 0	10ZN 13K 234K 10ZN 243N 230N 230N 230N 210ZN 244K 0 143N 23 N 238N 258N 381 3ZN 34ZN 230N 23 N 488 231K 30K 37K 37K 37K 37K 37K 37K 37K 37K 37K 37	9K 224	136K 16K 339K 158K 329K 333K 331K 331K 319K 25K 328K 330K 236K 340K 0	62K 11K 121K 75K 114K 122K 115K 105K 105K 18K 112K 124K 99K 177K 117K 0 44K 127K 26K 110K 116K 2K 125K	43K 540K 239K 23K 251K 230K 166K 235K 253K 258K	1K 241	119K 14K 249K 138K 242K 250K 237K 230K 226K 23K 230K 250K 179K 269K 251K 245K 57K 253K 45K	3K 8K	50K   130K   132K   131K   132K   126K   11K   133K   128K   63K   134K   132K   134K   17K   132K   20K   129K   131K   394	2K 245	67K 11K 151K 84K 146K 147K 144K 139K 135K 20K 139K 149K 110K 156K 144K 142K 49K 148K 26K 142K 146K 787	78K 28K 0 65K 24K 79K 75K 76K 78K 83K 7K 80K 75K 30K 72K 77K 79K 10K 76K 12K 77K 70 07V 07V 07V 07V 07V 07V 07V 07V 07V	5K 2K 4K	1K 83	3K 305	2K 220	10K 233K 177K 197K 241K 268K 3K 251K 162K 51K 148K 202K 222K 9K 210K 9K 210K 222K 129 162K	3K 116	0K 167	4K 249	19K 145	90K 12K 178K 107K 169K 182K 169K 168K 164K 28K 170K 180K 150K 180K 179K 174K 48K 185K 36K 166K 169K 7K 176K	32K 260	279K 68K 7K 276K 65K 268K 286K 286K 270K 9K 277K 275K 105K 265K 285K 285K 275K 275K 276K 195 265K	23K 225	6K 261	3K 161	0 171	0 12K 13K 10K 13K 15K 0 14K 11K 1K 8K 13K 10K 0 13K 12K 13K 12K 41K 150K 163K 165K 161K 15K 15K 15K 15K 15K 15K 15K 15K 15K 1	3K 11B	147K 20K 298K 182K 296K 307K 296K 2290K 41K 292K 296K 252K 304K 311K 301K 98K 308K 50K 292K 293K 8K 297K	7K 31C	31K 100K 105K 105K 101K 97K 37K 100K 105K 38K 105K 104K 101K 12K 102K 13K	0 57K 16K 101K 92K 76K 78K 136K 1K 120K 68K 111K 37K 95K 101K 3K 70K 2K 92K 103K 0 330K 47F 1M 640K 34K 70K 14K 70K 14K 70K 14K 70K 14K 70K 103K 00	3K 225	0K 205
en 6 19K 335K 2 188K 2	97K	10K 253K 1	95K 0	151K	142K			131K	200K	213K	319K 2	108K	239K 2	27K	226K	SK XX	126K	236K	135K 2	83K	SK S	87K	296K	214K	268K	116K	116K	240K 3	125K 2	164K	249K	270K	220K	251K 3	158K	21K	15K	13K	280K	316K	97K	136K	214K	203K
de         el         en         eo           11K         15K         19K         1K           342K         347K         335K         27K           201K         196K         188K         24K	79K 88K	10K K 259K	87K	X 157K	143K	7 87K	: 26K	137K 145K 138K 132K 131K 30K	7 185K	Z 226K	331K	K 109K	244K	21K	K 230K	N 342K	Z 132K	Z 248K	K 139K	76K 78K	4 ¥	1 81K	X 309K	C 228K	K 241K	K 116K	K 117K	Z 247K	K 150K	X 168K	K 282K	X 286K	K 223K	K 215K	K 163K	15K	12K 13K 10K 13K 15K 0	7 10K	X 290K	K 288K	X 101K	78K	C 221K	₹ 206K
a de K 11K 1K 342K 9K 201K	K 79K	K 9K 9K 269K	3K 73K	1091	0 X O O X O X X X X X X X X X X X X X X	K 94K	12K 22K 26K	5K 1381	7K 157	5K 2171	3K 3311	2K 115	JK 451	K 21K	9K 237	6K	3K 1311	JK 2471	7K 144	76F	K 5K	1K 73k	1K 296,	K 2141	7K 1971	3K 1111	ZK 128 K 1721	5K 2601	6K 158	2K 1691	OK 292.	5K 2861	5K 223	NK 274	3K 1671	K 17k	1K 10F	Z I K	7K 296	2K 216	5K 1031	16 76k	5K 2181	3K 207
cs da 11K 9K 342K 341K 204K 209K	85K 79K	10K 8K 260K 279K	89K 68K	26K 0	153K 160K	60K 9K 101K 67K 95K 98K 94K 87K	27K 12	37K 14	25K 135	20K 21c	29K 33.	14K 12	240K 250K	11K 19	42K 25	3K SO	30K 132	45K 249	46K 14	79K 75K	2K 5K	31K 70	04K 31	21K 22	33K 17.	15K 11.	25K 13. 71K 171	53K 260	31K 139	69K 18.	55K 136	58K 28	25K 22	30Z 21K	63K 168	5K 14	12K 13	8K 33	96K 30	3K 12	30K 10	01K 92	21K 21	01K 20.
bs ca cs 4K 0 11K 337K 182K 342K 206K 117K 204K	12K		0 149K	65K	78K	67K 9	0	97K 13	24K 18	62K 2	158K 3.	75K 1	123K 2	4K	138K 2	0 0	50K 1	22K 2.	84K 1	24K 7	40 0	3 X61	3/K 3	54K	10K	16K 1	73K 1	157K 2.	88 K 2 K 1 L	107K 1	133K 2	65K 2	106K 2	20K 2	68K 1,	0	0 41K	0	182K 2	45K 3	31K IA	16K 1.	67K 2.	41K 2
bs 4K 337K 206K	57K	<del>4</del> 0	52K	167K	163K	101K	10K	X 143K	103K	211K	339K	X 121K	X8K 193K 3	14K	< 249K	216	134K	197K	7 15 IK	65K	3K	56K	252K	71K 2K 220K	142K	93K	35K	125K 18K 265K 157K	K 156K	178K	100K	276K	X 224K	279K	175K	11K	10K	3K	298K	159K	105K	37K	215K	209K
bn br 2K 0 158K 17K 98K 15K	0	0 0 84K 5K	11K 5K	68K 7K 167K	108K 14K	0K 9K	0 2K	77K 13K	3C 3C	6K 7K	36K 16k	2K 11F	8K 2K	ў. О	19K 14k	N/C 0		0K 6K	7K 11k	28K 0			33K 0	IK 2K	66K 3K 142K	40K 0	60K 9K 135K 65K 8K 179K	25K 18K	6K 12F 9K 11K	0K 12k	33K 15k	8K 7K	2K 14k	3K 14K	3K 7K	1K 0	0 0 40K	χ. O	17K 20k	36K 3K		8K 0	5K 7K	0K 0
bg b 11K 2 347K 15			84K 11		236K 10	96K 66		142K 7	01 3/52	222K 86	332K 13	120K 6	246K 10	22K 8	245K 11	3K 0	33K 60	249K 30	149K 67	78K 28		82K 23	312K 69	226K 71	223K 66	118K 40	128K 66	258K 12	154K 76 135K 69	177K 90	288K 13	179K 68	220K 10	89K 15	67K 7.	17K 4	13K (	8K 2	300K 14		102K 41	78K 8	20K 9	205K 7A
ar 8K 0 211K	89K	9K 276K	82K	164K	155K	98K	20K	144K	ALC7	215K	338K 3		45K	24K		Acoc Acoc	132K	251K 2	145K	77K	2K	82K	312K	227K	244K		130K 179K	262K	157K 137K	184K	296K	278K	223K	302K 2	168K 1	20K	13K	8K	303K	271K	104K	90K	ZELII U 901N 913N 10N U 323N 47N 1M 1994N 301N 294N 1M 44N 120N 237N 308N 37N 308N 37N 308N 37N 308N 37N 308N 301N 30 301N 301N 301N 301N 301N 301N	196K
af 0 18K 20K	n 2K	0 X	0 21K	10K	15K	18K		22K	8K 0	5K	i 14K	13K	0 J	ΙX	hr 17K		¥	0	t 14K	2K	_	0	× ×	1 K	ml 5K 244K	IK	n 14K no 5K	20K	18K	ro 24K	ru 21K	9K	14K	sq 2K 218K	sv 7K	-	0 %	0	tr 28K	SK	3K	ze.en 2K	cn 8K	tw 4K
ar bg	툅	P S	3 3	ام!	8 =	′  =	60	8 1	<b>a</b>	g g	g	٠١٦	ᆲ	٦	- -	2 2	۱.=	.s	# .	e S	포	3,	= ≥	=	=	Sil 1	- -	ם	티	=	-   5	S .	8	ي   ي	s	=	੩ ਵ	=	۱۳۱	<b>#</b>	-	ze z	ş q	zh

Table 8: Bilingual lexicon counts for the entire word2word dataset. Each (row, column) entry is the number of words in the (row  $\rightarrow$  column) bilingual lexicon. See http: //opus.nlpl.eu/OpenSubtitles2018.php for language codes and original data sizes.