Harnessing Cross-lingual Features to Improve Cognate Detection for Low-resource Languages

Diptesh Kanojia[†,♣,⋆], **Raj Dabre**[◇], **Shubham Dewangan**[†],
Pushpak Bhattacharyya[†], **Gholamreza Haffari**[⋆], and **Malhar Kulkarni**[†]
[†]IIT Bombay, India, [◇]NICT, Japan
[♣]IITB-Monash Research Academy, India
[⋆]Monash University, Australia
[†]{diptesh,pb,malhar}@iitb.ac.in, [◇]raj.dabre@nict.go.jp
[†]sdofficial1996@gmail.com, [⋆]gholamreza.haffari@monash.edu

Abstract

Cognates are variants of the same lexical form across different languages; for example "fonema" in Spanish and "phoneme" in English are cognates, both of which mean "a unit of sound". The task of automatic detection of cognates among any two languages can help downstream NLP tasks such as Cross-lingual Information Retrieval, Computational Phylogenetics, and Machine Translation. In this paper, we demonstrate the use of cross-lingual word embeddings for detecting cognates among fourteen Indian Languages. Our approach introduces the use of context from a knowledge graph to generate improved feature representations for cognate detection. We then evaluate the impact of our cognate detection mechanism on neural machine translation (NMT), as a downstream task. We evaluate our methods to detect cognates on a challenging dataset of twelve Indian languages, namely, Sanskrit, Hindi, Assamese, Oriya, Kannada, Gujarati, Tamil, Telugu, Punjabi, Bengali, Marathi, and Malayalam. Additionally, we create evaluation datasets for two more Indian languages, Konkani and Nepali[1]. We observe an improvement of up to 18% points, in terms of F-score, for cognate detection. Furthermore, we observe that cognates extracted using our method help improve NMT quality by up to 2.76 BLEU. We also release[2] our code, newly constructed datasets and cross-lingual models publicly.

1 Introduction

India is a multilingual, multi-script country with 22 scheduled languages and 12 written script forms primarily belonging to 6 different language families. More than a billion people use these languages as their first language. A significant amount of news and information is found on the web in these languages, which is inaccessible to people of other regions within the country. Most of the Indian language texts found online have several words that have originated from Sanskrit, Persian, and English. While, in many cases, one might argue that such occurrences do not belong to an Indian language, the frequency of such usage indicates a wide acceptance of these foreign language words as Indian language words. In numerous cases, these words also are morphologically altered as per the Indian language morphological rules to generate new variants of existing words. Detection of such variants or 'Cognates' across languages helps Cross-lingual Information Retrieval (CLIR) (Makin et al., 2008; Meng et al., 2001), Machine Translation (MT) (Kondrak, 2005; Kondrak et al., 2003; Al-Onaizan et al., 1999), and Computational Phylogenetics (Rama et al., 2018). Cognates are etymologically related words across two languages (Crystal, 2011). However, NLP applications are concerned with the set of cognate words which have similarities in their spelling and their meaning. For example, the French and English word pair, *Liberté - Liberty*, reveals itself to be a true cognate through orthographic similarity. In some cases, similar words have a common meaning only in some contexts; such words are called partial cognates. For example, the word *"police"* in French can translate to *"police"*, *"policy"* or *"font"*, depending on the context[3]. Manual detection of such cognate sets requires a human expert with a good linguistic background in multiple languages. Moreover, manual annotation of cognate sets is a costly task in terms of time and human effort.

[1]It is primarily spoken in Nepal, but is also adopted in the list of scheduled languages of the Republic of India.

[2]Link: Data, code and models

[3]Cognates can also exist in the same language. Such word pairs/sets are commonly referred to as *doublets*.

Proceedings of the 28th International Conference on Computational Linguistics, pages 1384–1395
Barcelona, Spain (Online), December 8-13, 2020

The task of cognate detection across languages requires one to detect word pairs which are etymologically related, and carry the same meaning. Previous approaches to the task use orhtographic (Ciobanu and Dinu, 2014), phonetic (Rama, 2016) and semantic (Kondrak, 2001) features. However, these methods have a limitation since they do not take into consideration the notion of semantic similarity across languages. A key question that we try to answer in this paper is,

> *"Can semantic information be leveraged from Cross-lingual models to improve cognate detection amongst low-resource languages?"*

We hypothesize that utilizing cross-lingual features by employing existing resources such as wordnets and cross-lingual embeddings should help improve cognate detection. In this paper, we utilize the semantic information from cross-lingual word embeddings. Cross-lingual word embeddings can be obtained by training monolingual embeddings for individual languages and then projecting them in a shared space using a bilingual dictionary. In the absence of such a bilingual dictionary for low-resource languages, adversarial training can be used over identical words to generate the projections. We build cross-lingual models for thirteen language pairs with Hindi as the source (L1) and thirteen target Indian languages (L2). We use the context information from a knowledge graph to build the context dictionaries for each pair. The cross-lingual models help us obtain embeddings for the word-pair and the respective context dictionaries, from a shared space. We hypothesize that using this approach should provide a more accurate semantic measure for the detection of cognate pairs. The use of orthographic and phonetic similarity-based methods to perform the same task provides us with baselines for a comparative evaluation.

A motivation to investigate this task for low-resource Indian languages stems from the fact that most of the Indian languages borrow cognates or "loan words" from the Sanskrit language. It is, for the most part, considered a historical antecedent of almost all the Indian languages. Indo-Aryan languages like Hindi, Bengali, Gujarati, Punjabi borrow from Sanskrit. They borrow many lexical forms and language properties from Sanskrit. Dravidian languages are highly agglutinative and morphologically rich like Sanskrit, which makes them tough to parse computationally. Marathi and Hindi suffer from the same ailment even though Hindi is not considered as agglutinative as Marathi, but it does exhibit compounding[4] which makes it, yet again, difficult to parse for CLIR and MT systems, and to detect cognates based solely on orthographic similarity. Given that CLIR and MT are usually based on a full-form lexicon, one of the possible issues in the generation of cognates concerns the similarity of words in their root form vs the similarity in their lexical form. For example, the Sanskrit word "*matra*" and the English word "*Mother*" are known cognates from the Proto-Indo-European language family where the root and the meaning are identical, but the lexical form is considerably different. Our approach handles such cases by inculcating the sub-word information while building the embeddings and helps reduce the out-of-vocabulary (OOV) words, which have proven to be a challenge for well-established CLIR systems (Udupa et al., 2009).

This paper is organized as follows. Section 2 briefly describes the previous work done in the area of automatic cognate detection. Section 3 describes the dataset source, our additions to it, and the experimental setup. Section 4 presents the approaches used in terms of feature sets and classification methodologies. The results obtained are described in Section 5 along with a discussion on the qualitative analysis of our output. Section 6 concludes this article with possible future work in the area.

2 Related Work

The two main existing approaches for the detection of cognates belong to the *generative* and *discriminative* paradigms. The first set of approaches is based on the computation of a similarity score between potential candidate pairs. This score can be based on orthographic similarity (Jäger et al., 2017; Melamed, 1999; Mulloni and Pekar, 2006), phonetic similarity (Rama, 2016; List, 2012; Kondrak, 2000), or a distance measure with the scores learned from an existing parallel set (Mann and Yarowsky, 2001; Tiedemann, 1999). The discriminative paradigm uses standard approaches to machine learning, which are

[4]Compounding means when two or more words or signs are joined to make a longer word or sign.

based on (1) extracting features, *e.g.,* character n-grams, and (2) learning to predict the transformations of the source word needed to (Jiampojamarn et al., 2010; Frunza and Inkpen, 2009).

Cognate Detection has been explored vastly in terms of classification methodologies. Previously, Rama (2016) employ a Siamese convolutional neural network to learn the phonetic features jointly with language relatedness for cognate identification, which was achieved through phoneme encodings. Jäger et al. (2017) use SVM for phonetic alignment and perform cognate detection for various language families. Various works on orthographic cognate detection usually take alignment of substrings within classifiers like SVM (Ciobanu and Dinu, 2014; Ciobanu and Dinu, 2015) or HMM (Bhargava and Kondrak, 2009). Ciobanu and Dinu (2014) employ dynamic programming based methods for sequence alignment. Kanojia et al. (2019a) perform cognate detection for some Indian languages, but a prominent part of their work includes *manual verification and segratation* of their output into cognates and non-cognates. Kanojia et al. (2019b) utilize recurrent neural networks to harness the character sequence among cognates and non-cognates for Indian languages, but employ monolingual embeddings for the task. Dijkstra et al. (2010) show how cross-linguistic similarity of translation equivalents affects bilingual word recognition, even in tasks manually performed by humans. They discuss how the need for recognizing semantic similarity arises for non-identical cognates, based on the reaction time from human annotators. Similarly, Merlo and Andueza Rodriguez (2019) show that cross-lingual models exhibit the semantic properties of for bilingual lexicons despite their structural simplicities, which leads us to perform our investigation for low-resource Indian languages. Uban et al. (2019) discuss the semantic change in languages by studying the change in cognate words across Romance languages using cross-lingual similarity. All of the previous approaches discussed above, lack the use of an appropriate cross-lingual similarity-based measure and do not work well for Indian languages as shown in this work. This paper discusses the quantitative and qualitative results using our approach and then, applies our output to different neural machine translation architectures.

Language Pair	Hi-Bn	Hi-Gu	Hi-Mr	Hi-Pa	Hi-Sa	Hi-Ml	Hi-Ta	Hi-Te	Hi-As	Hi-Kn	Hi-Or	Hi-Ne*	Hi-Ko*
Cognates	15312	17021	15726	14097	21710	9235	3363	936	3478	4103	11894	2560	11295
Non-Cognates	16119	15057	15983	15166	23029	8976	4005	1084	4101	3810	13027	1918	9826

Table 1: Number of cognates and non-cognates for each language pair in the dataset. Hi-Ne* and Hi-Ko* were generated via replicating their approach (Kanojia et al., 2020).

Language	Hi	Bn	Gu	Mr	Pa	Sa	Ml	Ta	Te	Ne	As	Kn	Ko	Or
Corpus Size	48142K	1564K	439K	520K	505K	553K	495K	909K	1023K	706K	504K	159K	214K	744K
STTR (n=1000)	0.5821	0.5437	0.4587	0.6108	0.4314	0.5350	0.7339	0.6411	0.4950	0.4883	0.5968	0.5338	0.5614	0.4160

Table 2: Corpus Statistics where corpus size is the approximate number of lines, and STTR is the moving average type-token ratio on a windows of 1000 sentences.

3 Dataset and Experimental Setup

In this section, we describe our primary dataset for the cognate detection task. We also describe the datasets used for building cross-lingual word embedding models, and the parallel corpora used for the Neural Machine Translation (NMT). For our experiments, we use the publicly released challenge dataset (Kanojia et al., 2020) of cognates. This dataset provides labelled cognate and non-cognate pairs for twelve Indian languages namely, Sanskrit (Sa), Hindi (Hi), Assamese (As), Oriya (Or), Kannada (Kn), Gujarati (Gu), Tamil (Ta), Telugu (Te), Punjabi (Pa), Bengali (Bn), Marathi (Mr), and Malayalam (Ml). We reproduce their approach to add two more languages, Konkani (Ko) and Nepali (Ne), to this dataset. For building context dictionaries, we use linked Indian language wordnets (Bhattacharyya, 2017) and concatenate the concept

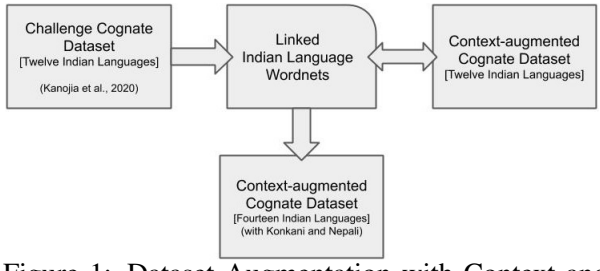

Figure 1: Dataset Augmentation with Context and Two Language Pairs using IndoWordnet.

definition and example sentences. We remove stop words from the context dictionaries and append them with their respective word pairs. The lexical overlap between the language pairs ranges from 13% (for Hi-Te) to only 23% (Hi-Mr). Figure 1 shows an accurate description of the dataset creation process. The cognate dataset statistics are described in Table 1.

Monolingual Corpora for Word Embeddings

The dataset for training cross-lingual models is obtained from various sources. Word embeddings require a large quantity of monolingual corpora for efficient training of a usable model with high accuracy. We extract corpora for these fourteen Indian Languages from various sources and collect them in a single repository. We extract Wikimedia dumps[5] for all languages and add Indian Language Corpora Initiative (ILCI) corpora (Jha, 2010) for these languages to each of them. For Hindi, Marathi, Nepali, Bengali, Tamil, and Gujarati we add crawled corpus of film reviews and news websites[6] to their corpus. For Hindi, we also add HinMonoCorp 0.5 (Bojar et al., 2014) to our corpus adding approximately 44 million sentences. For Sanskrit, we download a raw corpus of proses[7] and add it to our corpus. Training corpus statistics (approximate number of total lines) are shown in Table 2.

Parallel Corpora for NMT

To validate the application of cognates for the Machine Translation task, we choose the Neural Machine Translation setting and use the Indian Languages Corpora Initiative (ILCI) Phase 1 corpus. This corpus contains approximately 50K parallel sentences across 11 languages (English and 10 Indian Languages), from health and tourism domains. For every language pair, the parallel corpus was split up into a training set of 46,277 sentences, a test set of 2000 sentences and development set of 500 sentences. The train, test and development splits were ensured to be parallel across all language pairs involved. The language pair intersection for our cognate detection work and this parallel corpus limited our MT experimentation to the following languages namely, Hindi (Hi), Punjabi (Pa), Bengali (Bn), Gujarati (Gu), Marathi (Mr), Tamil (Ta), Telugu (Te) and Malayalam (Ml). We keep Hi as the source and remaining languages as the target languages for our experiments. We describe the experimental setup for our task below.

3.1 Unicode Offset based Transliteration

Indian languages use different scripts, and lexical similarity-based metrics cannot be directly used on any text for character matching. For standardization, we choose to convert any other script to the Devanagari script. We perform Unicode transliteration using Indic NLP Library[8] to convert scripts for Bn, As, Or, Gu, Pa, Ml, Ta, Kn and Te to Devanagari for standardization. Hi, Mr, Ko, Ne, and Sa are already based on the Devanagari script. We perform this for script transliteration for both the cognate dataset (Table 1) and the corpus (Table 2). We describe the creation of cross-lingual word embeddings below.

3.2 Cross-lingual Word Embedding Methodologies

Using the monolingual corpora described above, we build monolingual word embeddings using the FastText library[9] (Bojanowski et al., 2017) since it takes sub-word information into account, which is beneficial for a task such as ours where sub-words play an important role, and spelling variations can lead to different meanings. We do not use BERT (Devlin et al., 2018), ELMo (Peters et al., 2018), or M-BERT (Pires et al., 2019) for word embeddings as their pre-trained models are not trained on transliterated corpora. We choose FastText to train Skipgram word embedding models (100 dimensions) for each language using the following hyperparameters - 15 epochs with 0.1 as the learning rate. We use two characters (bi-gram) as the size of each sub-word for capturing the maximum number of sub-words.

We use three different methodologies for training the cross-lingual word embedding models on all the language pairs with Hindi as a pivot language (Hi-Mr, Hi-Bn and so on). The **first methodology**

[5]Link: Wikimedia Dumps; as on April 22, 2020
[6]Link: Additional Monolingual Corpus
[7]Link: JNU Sanskrit Proses Corpus
[8]Link: Indic NLP Library
[9]Link: FastText - GitHub

uses the supervised method named MUSE (Conneau et al., 2017)[10] which utilizes a manually curated bilingual lexicon[11] for alignments. We use Hindi as a pivot language due to the ease of computation and availability of resources (Corpora and WordNet size). We use the monolingual models described above and train 13 cross-lingual word embedding models (thirteen language pairs over 100 dimensions) using this approach.

The **second cross-lingual methodology** uses VecMap (Artetxe et al., 2018), which utilizes the monolingual models created above. VecMap uses an optional normalization feature while it builds the mappings between any two monolingual models. It performs orthogonal transformation and maps semantically related words, similar to MUSE, which was used in our first approach for building cross-lingual models. Additionally, it also reduces the dimensions of the embeddings models, which, is optional. We train it using the same hyperparameters as described above, for consistency while evaluating. We used the supervised approach for training these models as well, and the training dictionary was similar to the one provided to the MUSE method. We obtain thirteen models, one for each language pair, using VecMap. The **third methodology** utilizes contextual embeddings which have shown to outperform the conventional word embeddings based models for many tasks (Devlin et al., 2018). We choose the most recent methodology for building a single cross-lingual model for all the languages. XLM-R (Conneau et al., 2019) uses previously proposed approaches of XLM (Lample and Conneau, 2019) and RoBERTa (Liu et al., 2019) to attain a very high performing cross-lingual model, especially for low-resource languages. We use our transliterated corpora described above and concatenate it into a single large corpus required for training the model. We then use the unsupervised training method of XLM-R and train a model over six days and a couple more hours with a reduced batch size, which allowed us to train the model under a week's time. For this approach, however, we did not need a dictionary for the cross-lingual mapping strategy, unlike the two previous approaches.

To put it more concisely, we trained cross-lingual models using three different methodologies (MUSE, VecMap and XLM-R) where the cross-lingual mapping obtained for MUSE and VecMap were generated via the monolingual embeddings, as described above. We obtained thirteen models using each of these two methods. A single cross-lingual model was, however, trained using XLM-R and used for the third cross-lingual approach whose training methodology has been described above. We utilize the last layer from the XLM-R model to generate representations for each token.

4 Approaches

We use various approaches to perform the cognate detection task *viz.* baseline cognate detection approaches like orthographic similarity-based, phonetic similarity-based, phonetic vectors with Siamese-CNN based proposed by Rama (2016), and Recurrent neural network-based approach proposed by Kanojia et al. (2019b). We use the same hyperparameters and architectures, as discussed in these papers. We describe each of these feature sets in this section.

4.1 Weighted Lexical Similarity (WLS)

The Normalized Edit Distance (NED) approach computes the edit distance (Nerbonne and Heeringa, 1997) for all word pairs in our dataset. Each of the operations has unit cost (except that substitution of a character by itself has zero cost), so NED is equal to the minimum number of operations to transform 'word a' to 'word b'. We use a similarity score provided by NED, which is calculated as (1 - NED Score). We combine NED with q-gram distance (Shannon, 1948) for a better similarity score. The q-grams ('n-grams') are simply substrings of length q. This distance measure has been applied previously for various spelling correction approaches (Owolabi and McGregor, 1988; Kohonen, 1978). Kanojia et al. (2019b) propose this metric and we replicate it to generate features for their baseline approach. For any word pair with words p and q, it is as follows:

$$WLS_{pq} = (NED_{pq} * 0.75) + (QD_{pq} * 0.25) \quad (1)$$

[10]Link: MUSE - GitHub

[11]Link: Bilingual Lexicon

Now that this approach can be used to compute a score between each word pair, we use it to find two scores, which are used as features - 'word-pair similarity' and 'contextual similarity'. Each candidate word-pair generates a score *i.e.*, score1, and the average of scores among all words in the context dictionary generates another score *i.e.*, score2, which are normalized as follows:

$$\begin{aligned} S_1 &= score1/\left(score1 + score2\right) \\ S_2 &= score2/\left(score1 + score2\right) \end{aligned} \tag{2}$$

We use S_1 and S_2 as features for this orthographic similarity-based baseline approach.

4.2 Phonetic Vectors and Similarity (PVS)

The IndicNLP Library provides phonetic features based vector for each character in various Indian language scripts. We utilize this library to compute a feature vector for each word by computing an average over character vectors. We compute vectors for both words in the candidate cognate pairs (PV_S and PV_T) and also compute contextual vectors (PCV_S and PCV_T) by averaging the vectors for all the context dictionaries on each side (source and target), generating a total of four vectors. We also calculate the cosine similarity among PV_S and PV_T, and among PCV_S and PCV_T to generate two similarity scores (P_{S1}, and P_{S2}) which are normalized using (2) and, additionally, used as features during classification. It should be noted that using phonetic vectors and their similarity scores has already been proposed in the previous literature (Rama, 2016) for a cognate detection task, and we do not claim this approach to be our novel contribution.

4.3 Cross-lingual Vectors & Similarity

As described above, we train cross-lingual embedding models by aligning two disjoint monolingual vector spaces through linear transformations, using a small bilingual dictionary for supervision (Doval et al., 2018; Artetxe et al., 2017). The first two approaches for training cross-lingual methods use this dictionary for supervision. In our novel approach, we propose the use of vectors from the cross-lingual embedding models trained on Indian language pairs. We obtain vectors for word-pairs (WV_S and WV_T) and averaged context vectors (CV_S and CV_T) for the context dictionary, to create feature sets. We obtain vectors for each candidate pair and their context using all the three cross-lingual methodologies.

Additionally, we use angular cosine similarity (Cer et al., 2018) scores for word pairs and their contexts. Angular similarity distinguishes nearly parallel vectors much better as small changes in vector values yield considerable distances. For each word pair vector and its context vectors, we compute the 'word-pair similarity' and 'contextual similarity'. We use *arccos* to obtain angular cosine similarity (asim) among vectors 'u' and 'v', as shown below:

$$asim(u, v) = \left(1 - arccos\left(\frac{u.v}{\|u\|\|v\|}\right)/\pi\right) \tag{3}$$

Each candidate word-pair generates a score *i.e.*, score1, and the average of scores among all words in the context dictionary generates another score *i.e.*, score2, which are also normalized using (2).

4.4 Classification Methodology

We pose the task of detecting cognates as a binary classification problem. We employ both classical machine learning-based models and a simple feed-forward neural network. To compare our work with the previously proposed approaches, we replicate the best-reported systems from Rama (2016) *i.e.,* Siamese Convolutional Neural Network with phonetic vectors as features and also replicate Kanojia et al. (2019b)'s approach which uses a Recurrent Neural Network architecture with a weighted lexical similarity (WLS) as a feature set. The input to our classifiers is the feature sets described above for each candidate pair. The candidates are the complete data described in Table 1. Cognates from Table 1 are labelled positive, and non-cognates are labelled negative. *We perform 5-fold stratified cross-validation, which divides the data into train and test folds, randomly.* An architecture diagram for our classification approach is shown in Figure 2.

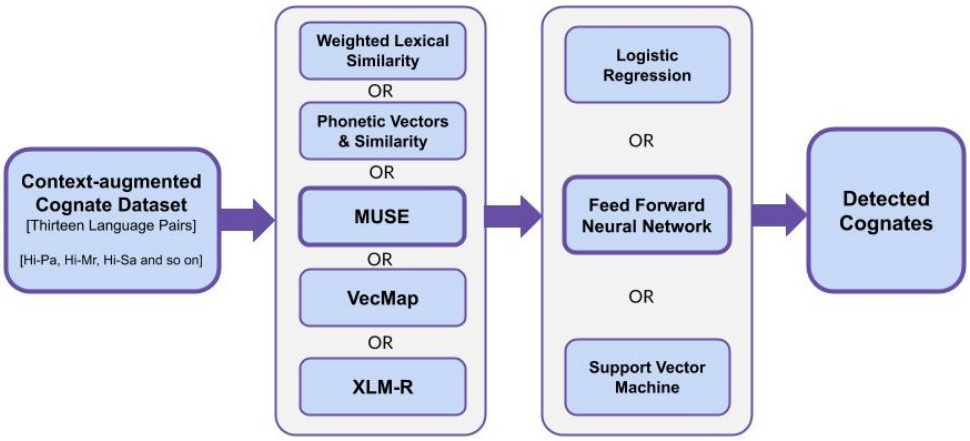

Figure 2: Cognate Detection task with different feature sets and classification approaches.

Among the classical machine learning models, we use Support Vector Machines (SVM) and Logistic Regression (LR). We experiment with the use of both linear SVMs and kernel SVMs (Gaussian and Polynomial). We perform a grid-search to find the best hyper-parameter value for C over the range of 0.01 to 1000. We deploy the Feed Forward Neural Network (FFNN) with one hidden layer. We perform cross-validation with different settings for activation function (tanh, hardtanh, sigmoid and relu) and the hidden layer dimension in the network (30, 50, 100, and 150). We use binary cross-entropy as the optimization algorithm. Finally, we choose the hyper-parameter configuration with the best validation accuracy. We train the model with the selected configuration with an initial learning rate of 0.4, and we halve the learning rate when the error on the validation split increases. We stop the training once the learning rate falls below 0.001. We perform our experiments with the feature sets (Orthographic (WLS), Phonetic (PVS), and three different cross-lingual embeddings based feature sets) described above for all the thirteen language pairs. We also perform an ablation test with various feature sets and report the results for the best feature combination in the next section. The results of our classification task can be seen in Table 3 and are discussed in the next section, in detail.

4.5 Cognate-aware Neural Machine Translation (NMT) Task

For the NMT task, we use the OpenNMT-Py toolkit (Klein et al., 2017) to perform our experiments. We use a Bidirectional RNN Encoder-Decoder architecture with attention (Bahdanau et al., 2014). We choose three stacked LSTM (Hochreiter and Schmidhuber, 1997) layers in the encoder and decoder. The hidden-size of the model was 500 units. We optimize using stochastic gradient descent at an initial learning rate of 1, and a batch-size of 1024 units. Training is done for 150,000 steps of which the initial 8,000 steps are for learning rate warm-up. We use Byte-pair encoding (BPE) (Sennrich et al., 2015) merge operations, initially, in an endeavour to find the best baseline model with an optimal number of merge operations. We observe that performing 2500 merge operations provided us with best BLEU (Papineni et al., 2002) scores, for most of the language pairs. We report the best results here, and a complete set of merge operation results in the supplementary material. We call this the NMT-BPE Baseline.

To validate our hypothesis that our approach can help the NMT task, we *inject the cognates detected using our approach* to the parallel corpus for their respective language pairs, as single word sentences. Lexical Dictionaries have previously been used to improve the MT task (Arthur et al., 2016; Han et al., 2019). However, a decent improvement in their BLEU scores is observed when their lexicon sizes are approximately around 1M tokens (Arthur et al., 2016). Our detected cognate list size varies from 930 cognates (Hi-Te) to 15834 (Hi-Mr). Due to the addition of more parallel instances to the corpus, the vocabulary size for NMT increases. Hence, we experiment further by varying the BPE merges, in a close range, to the optimal merge point obtained earlier. We report the results of the best optimal merge setting, for both NMT-BPE Baseline model and the cognate injected NMT-BPE model, in the section below. A more detailed set of results for all the merge operations is available in the supplementary material.

LP	Baseline Approaches									Cross-lingual Embeddings based Approaches									Best Combination		
	WLS w/ FFNN			PVS w/ Siamese CNN (Rama, 2016)			WLS w/ RNN (Kanojia et al., 2019)			XLM-R w/ FFNN			MUSE w/ FFNN			VecMap w/ FFNN			MUSE + WLS w/ FFNN		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Hi-Bn	0.51	0.28	0.36	0.68	0.62	0.65	0.67	0.69	0.68	0.81	0.76	**0.78**	0.77	0.75	0.76	0.72	0.74	0.73	0.80	0.75	0.77
Hi-As	0.48	0.26	0.34	0.72	0.71	0.71	0.72	0.70	0.71	0.70	0.72	0.71	0.80	0.75	**0.77**	0.74	0.73	0.73	0.84	0.75	**0.79**
Hi-Or	0.51	0.30	0.38	0.65	0.58	0.61	0.66	0.58	0.62	0.65	0.61	0.63	0.72	0.68	**0.70**	0.67	0.70	0.68	0.81	0.69	**0.75**
Hi-Gu	0.43	0.16	0.23	0.70	0.65	0.67	0.81	0.71	0.76	0.80	0.73	0.76	0.80	0.84	**0.82**	0.77	0.74	0.75	0.83	0.85	**0.84**
Hi-Ne	0.50	0.16	0.24	0.72	0.84	0.78	0.78	0.73	0.75	0.75	0.75	0.75	0.86	0.83	**0.84**	0.78	0.73	0.75	0.86	0.83	**0.84**
Hi-Mr	0.51	0.20	0.29	0.70	0.68	0.69	0.74	0.70	0.72	0.76	0.71	**0.73**	0.70	0.73	0.71	0.71	0.71	0.71	0.72	0.73	0.72
Hi-Ko	0.47	0.24	0.32	0.63	0.63	0.63	0.63	0.59	0.61	0.66	0.58	0.62	0.69	0.73	**0.71**	0.61	0.60	0.60	0.70	0.75	**0.72**
Hi-Pa	0.28	0.17	0.21	0.51	0.44	0.47	0.76	0.72	0.74	0.75	0.71	0.73	0.83	0.78	**0.80**	0.71	0.74	0.72	0.83	0.78	**0.80**
Hi-Sa	0.34	0.19	0.24	0.55	0.51	0.53	0.73	0.71	0.72	0.75	0.70	0.72	0.77	0.76	**0.76**	0.73	0.71	0.72	0.80	0.77	**0.78**
Hi-Ml	0.49	0.20	0.28	0.59	0.66	0.62	0.66	0.66	0.66	0.72	0.63	0.67	0.76	0.71	**0.73**	0.69	0.71	0.70	0.77	0.71	**0.74**
Hi-Ta	0.22	0.19	0.20	0.49	0.58	0.53	0.49	0.58	0.53	0.63	0.51	0.56	0.72	0.68	**0.70**	0.66	0.72	0.69	0.72	0.70	**0.71**
Hi-Te	0.18	0.15	0.16	0.60	0.71	0.65	0.62	0.71	0.66	0.65	0.70	0.67	0.70	0.72	**0.71**	0.67	0.67	0.67	0.73	0.72	**0.72**
Hi-Kn	0.19	0.18	0.18	0.54	0.60	0.57	0.58	0.60	0.59	0.60	0.58	0.59	0.69	0.73	**0.71**	0.65	0.64	0.64	0.70	0.73	**0.71**

Table 3: Results of the cognate detection task, in terms of weighted F-scores (5-fold) with baseline features and previous approaches, and our approaches using Cross-lingual similarity based features, for all the language pairs (LP).

5 Results and Discussion

From Table 3, among the baseline approaches, we observe high precision but very low recall scores when Weighted Lexical Similarity (WLS) based features are used. In fact, for language pairs which contain the Dravidian languages (Hi-Ml, Hi-Ta, Hi-Te, and Hi-Kn), even the precision scores are observed to be very low. The classifiers are not able to predict a significant amount of positively labelled cognate pairs, correctly. Even simple lexical variants such as "*Aag* (Fire)" (Hindi) and "*Agni* (Fire)" (Telugu) were classified incorrectly, as non-cognates. Phonetic vectors paired with a Siamese CNN (Rama, 2016), however, mitigate such misclassifications and are shown to perform well with much higher recall, for all the language pairs. Kanojia et al. (2019b)'s approach, however, outperforms the phonetic vectors based approach. We observe marginal improvements in F-scores for almost all the language pairs (except Hi-Ko and Hi-Ne) when their RNN based approach is used. As for our approaches, SVM and Logistic Regression based classification methodologies were consistently outperformed by the FFNN method. Hence, we report precision (P), recall (R), and F-scores (F) for only FFNN based approaches in Table 3.

Our cross-lingual similarity-based approaches, however, significantly outperform all the baseline approaches. We observe a stark improvement in both precision and recall scores for all the language pairs. The cross-lingual approach, which uses the vectors from VecMap based models, fails to outperform both MUSE and XLM-R based models. XLM-R model exclusively achieves the best f-score for two language pairs (Hi-Bn and Hi-Mr). We believe its performance can be attributed to the closeness of the language pairs as they belong to the same language family (Indo-Aryan). Moreover, XLM-R is a transformer architecture-based model which requires relatively larger corpora sizes and a decent amount of corpus was available to build word embedding models for these target languages (Table 2). The cross-lingual models built above are used to provide vectors for calculating the similarity between words and contexts, bringing in the notion of semantic similarity for the task of cognate detection. Please note that by the definition of cognates, they are semantically similar despite the lexical variance. We observe that MUSE based feature representations paired with FFNN, obtain the best F-scores. This observation stands true even when the target language belongs to the Dravidian language family, where our baseline approaches lack severely in performance. For example, "*mkarand-maKarantam* (pollen)" (Hi-Ta), a cognate pair was classified correctly only using the MUSE based approach.

Additionally, we perform an ablation test with our feature sets for further experimentation. We observe that the combination of WLS and vectors from the MUSE model performs even better. An improvement is observed for eight language pairs out of thirteen ranging from 1% point (Hi-Ko, Hi-Ml, Hi-Ta, Hi-

Te) to 5% points (Hi-Or). It should be noted that this is the only combination where no degradation in performance was observed for any language pair and hence, is reported in Table 3. Any other combination (MUSE + VecMap, MUSE + XLM-R, MUSE + PVS, and so on) degrades the performance of the cognate detection task, on at least one language pair.

The average improvement observed by using our best model (MUSE + WLS) over the strongest baseline approach (Kanojia et al., 2019b) is 9% points with the highest being 18% points (Hi-Ta). Over the weakest baseline approach (WLS), our best model obtains an average improvement of 50%, peaking at 61% points (Hi-Or).

Approaches / LP	Hi-Pa	Hi-Bn	Hi-Gu	Hi-Mr	Hi-Ta	Hi-Te	Hi-Ml
NMT-BPE Baseline	62.79	28.75	52.17	31.66	13.78	19.18	10.4
Cognate-aware NMT-BPE	**65.55**	**29.43**	**52.39**	**32.41**	**13.85**	**19.58**	**11.18**

Table 4: Results of the Cognate-aware Neural Machine Translation Task, in terms of BLEU scores, for the language pairs (LP) with available parallel data.

We present the results of Cognate-aware NMT in Table 4. For the Hi-Pa language pair, an improvement of 2.76 BLEU is observed, where 15001 cognates were detected including the misclassified pairs. Amongst a consistent improvement for all the language pairs, even when 930 cognate pairs (Hi-Te) are added, an improvement of 0.4 BLEU can be seen. The maximum number of cognate pairs injected into the NMT pipeline is 15834 pairs for the Hi-Mr language pair. Surprisingly, we do not observe the most significant improvement for Hi-Mr despite the largest number of cognates injected. We believe that this is because Marathi is a morphologically rich language which exhibits agglutination.

6 Conclusion and Future Work

In this paper, we harness cross-lingual embeddings to improve the task of cognate detection for thirteen Indian language pairs. We propose the use of a linked knowledge graph to augment a publicly released cognate dataset with a context dictionary. We reproduce the proposed approach and add two additional language pairs to the same dataset and perform experiments using various approaches for a comparative evaluation. We reproduce the previously proposed approaches (Rama, 2016; Kanojia et al., 2019b) for this task to perform a further evaluation. We obtain monolingual Indian language corpora for all the fourteen languages (Section 3), from various sources to build monolingual models and use a bilingual dictionary to supervise the task of cross-lingual models generation (MUSE and VecMap), for thirteen language pairs (Hi-Mr, Hi-Ta and so on). We also train a single cross-lingual model using the contextual embedding based approach (XLM-R).

Our experiments use three different approaches to generate better feature representations for the cognate detection task, and all of them show improvements over previously proposed approaches. We observe consistent improvements in terms of precision, recall and F-scores. We also perform an ablation study and show that augmenting WLS baseline feature with MUSE based features provide us with the best results. Over the strongest baseline, this model shows improvements up to 18% points, in terms of F-score. Our best F-score is observed for the Hi-Gu and Hi-Ne language pairs (0.84) which can still be improved and warrants further investigation into the task. Additionally, we use the detected cognate pairs and use a simple approach to inject them into the neural machine translation pipeline. Our Cognate-aware NMT-BPE results also show a consistent improvement for all the Indian language pairs. Furthermore, we release this augmented dataset, along with our code and cross-lingual models for further research.

In future, we aim further to investigate the performance of contextual embeddings for this task. Recent trends show that contextual embeddings based models outperform conventional word embeddings for most tasks. We, however, do not observe this and attribute this primarily due to the dataset size used to train the contextual embeddings. We shall add more data to our monolingual corpora and perform more experiments using XLM-R. Future experiments with cognate-aware NMT using the Transformer architecture (Vaswani et al., 2017) should further help in showing the importance of our extracted cognate pairs. We also aim to investigate the task of cognate detection for other Indian language pairs, along with Indo-European language pairs, in the near future.

Acknowledgements

We thank the lexicographers and annotators at the CFILT Lab, IIT Bombay for their efforts in creating the dataset for this study. We acknowledge the computational resources provided by NLP Lab at Monash University, and CFILT, IIT Bombay for performing the experiments. We also thank all the reviewers for their critique, which helped us shape up the article.

References

Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Franz-Josef Och, David Purdy, Noah A Smith, and David Yarowsky. 1999. Statistical machine translation. In *Final Report, JHU Summer Workshop*, volume 30.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada, July. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 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.

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. *arXiv preprint arXiv:1606.02006*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.

Aditya Bhargava and Grzegorz Kondrak. 2009. Multiple word alignment with profile hidden markov models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Student Research Workshop and Doctoral Consortium*, pages 43–48. Association for Computational Linguistics.

Pushpak Bhattacharyya. 2017. Indowordnet. In *The WordNet in Indian Languages*, pages 1–18. Springer.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ondrej Bojar, Vojtech Diatka, Pavel Rychlỳ, Pavel Stranák, Vít Suchomel, Ales Tamchyna, and Daniel Zeman. 2014. Hindencorp-hindi-english and hindi-only corpus for machine translation. In *LREC*, pages 3550–3555.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Alina Maria Ciobanu and Liviu P Dinu. 2014. Automatic detection of cognates using orthographic alignment. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 99–105.

Alina Maria Ciobanu and Liviu P Dinu. 2015. Automatic discrimination between cognates and borrowings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 431–437.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

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. *arXiv preprint arXiv:1911.02116*.

David Crystal. 2011. *A dictionary of linguistics and phonetics*, volume 30. John Wiley & Sons.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ton Dijkstra, Koji Miwa, Bianca Brummelhuis, Maya Sappelli, and Harald Baayen. 2010. How cross-language similarity and task demands affect cognate recognition. *Journal of Memory and language*, 62(3):284–301.

Yerai Doval, Jose Camacho-Collados, Luis Espinosa-Anke, and Steven Schockaert. 2018. Improving cross-lingual word embeddings by meeting in the middle. *arXiv preprint arXiv:1808.08780*.

Oana Frunza and Diana Inkpen. 2009. Identification and disambiguation of cognates, false friends, and partial cognates using machine learning techniques. *International Journal of Linguistics*, 1(1):1–37.

Dong Han, Junhui Li, Yachao Li, Min Zhang, and Guodong Zhou. 2019. Explicitly modeling word translations in neural machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(1):1–17.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.

Gerhard Jäger, Johann-Mattis List, and Pavel Sofroniev. 2017. Using support vector machines and state-of-the-art algorithms for phonetic alignment to identify cognates in multi-lingual wordlists. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1205–1216.

Girish Nath Jha. 2010. The TDIL program and the Indian langauge corpora intitiative (ILCI). In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May. European Language Resources Association (ELRA).

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2010. Integrating joint n-gram features into a discriminative training framework. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 697–700. Association for Computational Linguistics.

Diptesh Kanojia, Malhar Kulkarni, Pushpak Bhattacharyya, and Gholemreza Haffari. 2019a. Cognate identification to improve phylogenetic trees for indian languages. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, pages 297–300. ACM.

Diptesh Kanojia, Kevin Patel, Pushpak Bhattacharyya, Malhar Kulkarni, and Gholemreza Haffari. 2019b. Utilizing wordnets for cognate detection among indian languages. In *Global Wordnet Conference (2019)*.

Diptesh Kanojia, Malhar Kulkarni, Pushpak Bhattacharyya, and Gholamreza Haffari. 2020. Challenge dataset of cognates and false friend pairs from indian languages. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3096–3102.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Teuvo Kohonen. 1978. A very fast associative method for the recognition and correction of misspelt words, based on redundant hash addressing. In *Proceedings of the fourth International Joint Conference on Pattern Recognition, 1978*.

Grzegorz Kondrak, Daniel Marcu, and Kevin Knight. 2003. Cognates can improve statistical translation models. In *Companion Volume of the Proceedings of HLT-NAACL 2003-Short Papers*, pages 46–48.

Grzegorz Kondrak. 2000. A new algorithm for the alignment of phonetic sequences. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 288–295. Association for Computational Linguistics.

Grzegorz Kondrak. 2001. Identifying cognates by phonetic and semantic similarity. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Grzegorz Kondrak. 2005. Cognates and word alignment in bitexts. *Proceedings of the tenth machine translation summit (mt summit x)*, pages 305–312.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Johann-Mattis List. 2012. Lexstat: Automatic detection of cognates in multilingual wordlists. In *Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH*, pages 117–125. Association for Computational Linguistics.

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. *arXiv preprint arXiv:1907.11692*.

Ranbeer Makin, Nikita Pandey, Prasad Pingali, and Vasudeva Varma. 2008. Experiments in cross-lingual ir among indian languages. *Advances in Multilingual and Multimodal Information Retrieval. Springer Berlin/Heidelberg*.

Gideon S Mann and David Yarowsky. 2001. Multipath translation lexicon induction via bridge languages. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics.

I Dan Melamed. 1999. Bitext maps and alignment via pattern recognition. *Computational Linguistics*, 25(1):107–130.

Helen M Meng, Wai-Kit Lo, Berlin Chen, and Karen Tang. 2001. Generating phonetic cognates to handle named entities in english-chinese cross-language spoken document retrieval. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2001. ASRU'01.*, pages 311–314. IEEE.

Paola Merlo and Maria Andueza Rodriguez. 2019. Cross-lingual word embeddings and the structure of the human bilingual lexicon. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 110–120, Hong Kong, China, November. Association for Computational Linguistics.

Andrea Mulloni and Viktor Pekar. 2006. Automatic detection of orthographics cues for cognate recognition. In *LREC*, pages 2387–2390.

John Nerbonne and Wilbert Heeringa. 1997. Measuring dialect distance phonetically. In *Computational Phonology: Third Meeting of the ACL Special Interest Group in Computational Phonology*.

Olumide Owolabi and DR McGregor. 1988. Fast approximate string matching. *Software: Practice and Experience*, 18(4):387–393.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy, July. Association for Computational Linguistics.

Taraka Rama, Johann-Mattis List, Johannes Wahle, and Gerhard Jäger. 2018. Are automatic methods for cognate detection good enough for phylogenetic reconstruction in historical linguistics? *arXiv preprint arXiv:1804.05416*.

Taraka Rama. 2016. Siamese convolutional networks for cognate identification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1018–1027.

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

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Jorg Tiedemann. 1999. Automatic construction of weighted string similarity measures. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Ana Uban, Alina Maria Ciobanu, and Liviu P. Dinu. 2019. Studying laws of semantic divergence across languages using cognate sets. In *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change*, pages 161–166, Florence, Italy, August. Association for Computational Linguistics.

Raghavendra Udupa, K Saravanan, Anton Bakalov, and Abhijit Bhole. 2009. "they are out there, if you know where to look": Mining transliterations of oov query terms for cross-language information retrieval. In *European Conference on Information Retrieval*, pages 437–448. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.