Subword-based Cross-lingual Transfer of Embeddings from Hindi to Marathi

Anonymous EMNLP submission

Abstract

Embeddings are growing to be a crucial resource in the field of NLP for any language. This work focuses on subword embeddings transfer for Indian languages from a relatively higher resource language to a genealogically related low resource language. We work with Hindi-Marathi as our language pair, simulating a low-resource scenario for Marathi. We demonstrate the consistent benefits of unsupervised morphemic segmentation on both source and target sides over the treatment performed by FastText. We show that a trivial "copy-and-paste" embeddings transfer based on even perfect bilingual lexicons is inadequate in capturing language-specific relationships. Finally, our best-performing approach uses an EM-style approach to learning bilingual subword embeddings; the resulting embeddings are evaluated using the publicly available Marathi Word Similarity task as well as Wordnet-Based Synonymy Tests. We find that our approach significantly outperforms the FastText baseline on both tasks; on the former task, its performance is close to that of pretrained FastText Marathi embeddings that use two orders of magnitude more Marathi data.

1 Introduction

Subword-level embeddings are useful for many tasks, but require large amounts of monolingual data to train. While about 14 Indian languages such as Hindi, Bengali, Tamil, and Marathi have the required magnitudes of data and resources, most Indian languages are highly under-resourced; they have very little monolingual data and almost no parallel data, low internet presence and not much digitization. For example, to the best of our knowledge, Marwadi, spoken by 14M people, has no available monolingual corpus; Konkani, spoken by about 3M people, has a monolingual corpus containing 3M tokens, and no parallel data.[1] However, many of these languages have very close syntactic, morphological, and lexical connections to surrounding languages including the mentioned high-resource languages. Our approach aims to leverage these connections in order to build embeddings for these low-resource languages, in the hope that this will aid further development of other NLP tools such as MT or speech tools for them.[2]

In this study, we work with Hindi-Marathi as our language pair, and use asymmetric resources (large data for Hindi, artificially small monolingual data for Marathi) in line with the assumption of a low-resource language that is genealogically and culturally related to a high-resource language. We are constrained, of course, by the necessity of evaluation datasets for the resulting embeddings. This is intended to be a pilot work to a broader study that applies and perhaps adapts our given approach to a much larger coverage of different typologies of Indian languages and language pairs, in the hope of making it generalizable to truly low-resource languages. In this work, however, we only report its performance with respect to the Marathi bilingual embeddings obtained in our simulation of the low-resource setting.

Several Indian languages are morphologically rich, including Hindi and Marathi. This means that while related language pairs may have a high number of cognates, these may be "disguised" by surrounding inflectional or derivational morphemes. In this situation, even with an identical underlying syntactic structure, lexical correspondences between languages may be obscured or rendered incongruent. Further, when working with small data, the corpus frequencies of fully inflected surface forms would be much less reliable than those of stem and affix morphemes, intuitively resulting in

[1] The Opus Corpus (Tiedemann, 2012) is perhaps one of the most popular collection of parallel texts and can be checked for estimation of parallel data.

[2] While some languages may have a little parallel data, we assume none, so as to cater to languages that are just undergoing digitization, such as Mundari (has a monolingual corpus of 0.5M tokens and no parallel data).

a less robust embeddings transfer. These factors, along with the intuition that many Indian languages share morpheme-level correspondences with each other, motivated us to train and apply unsupervised morphemic segmentation on both the source and target language data; we demonstrate the benefits of doing so in our evaluations. The idea of the transfer is to project the low-resource language (LRL) subwords into a shared bilingual space with the high-resource language (HRL). We first attempt a trivial transfer that simply finds the "closest" HRL subword for each LRL subword, and copies its embedding. We demonstrate that this approach, while tempting, is not enough to capture the relationships between even identical words in both languages; embeddings spaces appear to encode more complex information that this approach would suggest. For our best performing approach, we use the EM-style algorithm described in Artetxe et al. (2017), which alternately optimizes the distance between pairs belonging to a bilingual mapping, and generates a bilingual mapping between words from the resulting bilingual embeddings. We compare the resulting Marathi bilingual embeddings to a FastText model trained on the available data as well as pretrained models, on the Word Similarity tasks and the Wordnet-Based Synonymy Tests.

Section 2 describes previous relevant literature, Section 3 describes some key properties of Hindi and Marathi, Section 4 records the datasets and tools we have used. Sections 5 and 6 describe our segmentation efforts and experiments respectively, and Sections 7 and 8 describe our results. We discuss our findings in Section 9, and conclude with potential directions of future research in Section 10.

2 Previous Work

2.1 Subwords in Embedding Spaces

In a seminal work, Bojanowski et al. (2017) present FastText, an algorithm to train embeddings that treats morphology by representing words as bags of chargrams. This idea has since been taken forwards by many studies that recognize its usefulness for morphologically rich languages as well as in data scarcity situations; Zhu et al. (2019b) look at the segmentation of a word, such as using chargrams, Byte Pair Encoding (BPE) (Gage, 1994; Sennrich et al., 2016), Morfessor, as well as the composition of the subword embeddings (addition, averaging, etc.) to construct the final word vector, and conclude that the best performing configuration is highly language and task dependent. A subsequent work (Zhu et al., 2019a) focuses on LRLs and finds the combination of BPE and addition largely robust, although they once again note language-dependent variability. They also find that encoding "affix" information with positional embeddings is beneficial, hinting that the embedding space may distinguish the importance of different kinds of subwords.

2.2 Cross lingual embeddings

The problem of learning bilingual embeddings has usually been studied in a symmetric resources scenario. Xu et al. (2018) propose a method of mapping two sets of monolingual embeddings into a shared space using gold bilingual lexicons; they present results for English paired with Spanish, Chinese, and French, evaluated on the bilingual lexicon induction as well as Word Similarity tasks. Chaudhary et al. (2018) work with training bilingual subword embeddings for pairs of Indian LRLs from scratch; they project different scripts into the International Phonetic Alphabet (IPA) and find that a joint learning objective performs better than transfer learning. Kayi et al. (2020) present an extension of the BiSkip cross lingual learning objective that leverages subword information to train English-paired bilingual embeddings for LRLs; however, they assume the presence of around 30K parallel sentences. We describe Artetxe et al. (2017) in some detail below, since we use this algorithm in our approach.

2.3 Bilingual Lexicon Induction

This task is closely related to that of embeddings transfer, especially when the languages under question are have many lexical correspondences. We therefore see that these two tasks often leverage each other; Hauer et al. (2017) uses word2vec embeddings (Mikolov et al., 2013) in order to iteratively train a translation matrix. However, older works such as Koehn and Knight (2002) and Haghighi et al. (2008) use monolingual features such as frequency heuristics, orthographic features, tags, and context vectors in order to find bilingual mappings.

2.4 Summarizing Artetxe et al. (2017)

This work presents an EM-style approach to training bilingual embeddings from monolingual embeddings without parallel data; however, it assumes

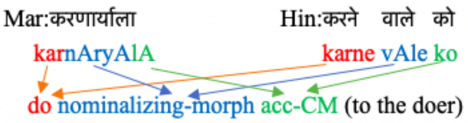

Figure 1: Tokens (with transliterations) in Marathi and Hindi. The stem for "do" is the same (i.e. "kar") in both languages; Marathi uses one token whereas Hindi uses three.

high quality monolingual embeddings for both languages trained on at least 1 billion word corpora each. Given the two sets of word embeddings, they find a bilingual dictionary D by choosing the closest target word for each source word with respect to the cosine distance between source and target word embeddings. In the next step, they use the dictionary D to calculate a linear transformation matrix that minimizes the sum of cosine distances of the embeddings of all word pairs in D. They apply an orthogonality constraint on the transformation matrix in order to preserve monolingual invariance i.e. to prevent the degradation of the monolingual relationships in the resulting embedding space.

3 Note on languages

Hindi is spoken by about 340M people.[3] It is related to other large Indian languages such as Marathi, Punjabi, and Bangla, and has 48 recognized "dialects" over India, which makes it a good choice for the HRL in this project. Hindi is written in the Devanagari script, which is also used for over 120 other (often related) languages, including Marathi. Both Hindi and Marathi are largely free word order; nouns inflect for case and number, verbs inflects for tense, number, gender, and adjectives inflect for gender, case, and number. Some differences are that Marathi exhibits more agglutinative tendencies than Hindi; Marathi is conventionally written in a manner that allows suffix stacking with certain boundary changes. For example, a Marathi token may be a sequence of verb+nominalizing-morpheme+case-marker or noun+postposition+genitive. In Hindi, while the morpheme order and inflection would largely be the same, these morphemes (such as case-markers, postpositions, and genitives) would appear as separate tokens (see Figure 1).

4 Data and Resources

4.1 Training Data

For Hindi, we used 1M sentences containing roughly 18M tokens from the HindMonoCorp 0.5 (Bojar et al., 2014). For Marathi, we used 50K sentences containing 0.8M tokens from the IndicCorp Marathi monolingual dataset (Kakwani et al., 2020). The latter number was chosen because it seems to be the ballpark of the amount of monolingual data collected for newly digitized languages.[4]

4.2 Pretrained Embeddings

We use pretrained FastText embeddings for Hindi, presented by Grave et al. (2018), in line with the assumption that we have good quality resources for the HRL. These embeddings are trained on the *Wikipedia* corpus as well as the *Common Crawl*, containing a total of about 2G tokens. We also use the pretrained Marathi FastText embeddings presented in the same work, solely for the purpose of comparison and evaluation; these embeddings are trained on 50M tokens and 85K Wikipedia articles.

4.3 Evaluation datasets

4.3.1 Word Similarity Dataset

A Word Similarity dataset is a set of word pairs, each annotated by humans according to the degree of similarity (on a scale of 1-10) between the two words. For evaluation, we find the cosine similarity between the two words, and ultimately calculate the Spearman's Rank Correlation between the human and model "similarity" judgments for all word pairs. We report this correlation multiplied by 100.

We present results on the Marathi Word Similarity dataset presented by Akhtar et al. (2017), containing 104 word pairs, as our primary evaluation. This dataset is created by translating a subset of the WordSimilarity-353 English evaluation dataset into Marathi by native Marathi speakers fluent in English, and re-evaluating the similarity scores by 8 native speaker annotators.

4.3.2 Wordnet-Based Synonymy Tests

Since the WordSim dataset is rather small, we also perform Wordnet-Based Synonymy Tests (WBST) (Piasecki et al., 2018). A WBST consists of a set of "questions" of the following format: there is a

[3] See https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers

[4] See https://www.ldcil.org/resourcesTextCorp.aspx for efforts on collecting data on under-resourced languages such as Bodo, Dogri, Santhali, etc.

"query word", and N options. One of the options is a synonym or closely related to the query word, while the rest are "detractors", or randomly selected words. The task is to identify the synonym; we do this by calculating the distances between the query word and each of the options in the embedding space and selecting the closest. The reported score is the percentage of questions that the embeddings answered correctly. We use the Marathi Wordnet,[5] containing 32K words, built by Sinha et al. (2006); Debasri et al. (2002), for generating the WBST. We also use the Python application interface given by Panjwani et al. (2018). Note that we use the Wordnet solely for evaluation purposes.

5 Segmentation

Due to the fusional/agglutinative nature of the languages, as well as the differences in morphology and tokenization conventions as discussed in Section 3, we apply morphemic segmentation to both source and target side data. This is motivated by the need to handle data scarcity on the LRL side, since fully inflected tokens are much rarer than their constituent subwords; we see that the unsegmented Marathi data has 100K distinct tokens, but only 20K distinct "morphemes" post-segmentation. In principle, the morphemic segmentation is also an attempt to increase the visibility of the morphs in the language data since it is easier to find correspondences between the two languages at this level rather than at the token level.[6] This is clear in the fact that 50% of the "morphs" in the Marathi segmented data also occur in the Hindi corpus, whereas for the unsegmented data, this is only 20% of tokens. We experimented with BPE and Morfessor and decided to use the latter, since BPE seemed unable to preserve longer morphs regardless of parameter settings. However, this can only be decision may vary according to language type.[7]

6 Approach

Our experiments test different intuitions about the cross-lingual interactions between the languages in question. As a baseline, we train a FastText model with 300 dimensions on the tokenized Marathi data with 0.8M tokens (Mar-Base-300-0.8M).

6.1 Normalized Edit Distance

This approach is based on a bilingual mapping at a subword level; it takes advantage of the high number of cognates between related languages as well as the common script. Its primary intuition is that since the languages share not only cognates but also syntactic and morphological properties, embedding vectors can essentially be "copied" over to the LRL from the HRL.

For each Marathi morph, we choose the Hindi morph with the minimum normalized edit distance from it. NED is calculated in the following way:

$$NED(l,h) = \frac{edit\_distance(l,h)}{max(length(l), length(h))}$$

To obtain the embedding of any Marathi word, we first segment it. For each subword, we

- Look for the closest Hindi morph by NED
- Retrieve the corresponding Hindi subword embedding

Finally, we compose the subword embeddings in any way we choose (such as addition).[8] See Algorithm 1 for a depiction.

Algorithm 1: Self/NED Mapping

```
l_word ← LRL word;
H_EMB ← HRL embeddings;
l_morphs ← $segment\_lrl$(l_word);
  l_subwords_emb ← empty list
for $l\_morph$ in $l\_morphs$ do
    h_closest ← $closest\_HRL\_morph$(l_word);
    $append$(l_subwords_emb, H_EMB(h_closest));
end
l_emb ← $compose\_subwords$(l_subwords_emb);
return l_emb ;
```

6.2 Self-learning Bilingual Lexicon approach

The approach proposed by Artetxe et al. (2017) in intended to generate bilingual *word* embeddings for equally well-resourced languages (See Section 2.4 for details). We hypothesize that it will maintain its quality at the subword level for morphologically rich languages; further, we hypothesize that in our data-asymmetry situation, this approach will serve to "transfer" some of the higher quality of

[5] See http://www.cfilt.iitb.ac.in/wordnet/webmwn/

[6] According to native bilingual speaker intuition

[7] Note that we do not lemmatize our data; good-quality lemmatizers are a scarce resource that we cannot assume for the LRL. Furthermore, we want to train subword embeddings for non-root morphemes as well and so we retain them in the data.

[8] Of course, this approach is highly limited to related words in the language. However, testing it out gives us an interesting insight about cognates and identical words (see Section 9.1)

the HRL embedding space to the LRL embeddings, by leveraging a bilingual mapping to induce the relationships already encoded in the HRL embeddings. We use 300-dimensional FastText vectors trained on available Marathi segmented data (Mar-Seg-300-0.8M) as the initial set of LRL embeddings. For the HRL, we can use any available resource. We try using pretrained FastText vectors (Hin-Pret-300-2G); we also retrain FastText on the segmented Hindi data (Hin-Seg-300-18M). For all runs, we set the initial seed dictionary as identical words in the source and target corpora.[9] We use the Mar-Seg-300-0.8M FastText model as a backoff for unseen morphs, as shown in Algorithm 2. For composing the subword embeddings of a word, we tried addition, averaging, and also simply picking the first subword embeddings and discarding the rest. The idea behind the last one is that this approximates the stem of the word, and also reduces the noise created by summing different subword embeddings.

Algorithm 2: Using bilingual embeddings with backoff

```
l_word ← LRL word;
L_EMB ← Bilinual LRL embeddings;
L_EMB_backup ← Mar-Seg-300;
l_morphs ← $segment\_lrl$(l_word);
l_subwords_emb ← empty list;
for l_morph in l_morphs do
    l_morph_emb ← empty list ;
    if l_morph in L_EMB then
        l_morph_emb ← L_EMB(l_word);
    end
    else
        l_morph_emb←L_EMB_backup(l_morph);
    end
    $append$(l_subwords_emb, l_morph_emb);
end
l_emb← $compose\_subwords$(l_subwords_emb);
return l_emb ;
```

7 Results: Word Similarity Task

All results are evaluated on the Marathi Word Similarity dataset as explained in Section 4.3.1.

7.1 Baseline and Comparison Models

We show the performance of Mar-Base-300-0.8M and Mar-Seg-300-0.8M; taking motivation from Chaudhary et al. (2018), we also try a joint approach i.e. we train bilingual embeddings jointly on the segmented Hindi and Marathi data (Joint-Seg-300-0.8M-18M). See Table 1 for these scores.

Approach	Score
Mar-Base-300-0.8M	24.64
Mar-Seg-100-0.8M	39.19
Mar-Seg-300-0.8M	**43.23**
Joint-Seg-300-0.8M-18M	35.48

Table 1: Marathi Monolingual and Marathi-Hindi Joint results on WordSim task. In all monolingual models, the final suffix refers to the number of tokens used for training; in all bilingual models, the last two suffixes refer to the Marathi and Hindi tokens, respectively, that were used in the approach.

Embeddings	Score
Mar-Pret-300-60M	**54.89**
Mar-SG-300-27M	41.12
Hin-Pret-300-2G	39.94

Table 2: Scores of high-resource Marathi and Hindi models on Marathi WordSim task for comparison.

In Table 2, we show the performance of pretrained FastText Marathi embeddings mentioned in Section 4.2 (Mar-Pret-300-60M), as well as the best performing model score from Akhtar et al. (2017) on this evaluation dataset. Akhtar et al. (2017) test different sets of embeddings including Skip-gram, CBOW (Mikolov et al., 2013) and FastText (Bojanowski et al., 2017) algorithms, all trained on a corpus with 27M tokens, of which the Skip-Gram (Mar-SG-300-27M) performed best.

From Table 1, we observe that simple segmentation of the data causes an improvement of over 20 points, outdoing not only Mar-Base-300-0.8M but Mar-SG-300-27M (See Table 2). Surprisingly, the joint model Joint-Seg-300-0.8M-18M dips in performance in comparison to the Mar-Seg-300-0.8M. We discuss this effect of the Hindi data on the bilingual embeddings in Section 9.1. Finally, Table 3 shows the performance of the Mar-Pret-300-60M and Hin-Pret-300-2G on certain word pairs in the Marathi WordSim dataset such that both words are also used identically in Hindi.[10] These word pairs were manually identified from the Marathi evaluation dataset; we found that there were 64 such

[9] Note that this approach does not use any parallel data or bilingual lexicons; this aligns with our assumptions about parallel data. However, in the case that parallel data does exist, it can be used to find a good quality bilingual seed lexicon in lieu of using identical words; this has been shown to improve the quality of the resulting bilingual embeddings.

[10] That is, both of the words in the word pair must be both Hindi and Marathi words with the same spelling, and near-identical senses.

Embeddings	Identical Word Score
Hin-Pret-300-2G	41.17
Mar-Pret-300-60M	**50.38**

Table 3: Hindi and Marathi embeddings scores for words from the Marathi WordSim dataset that are identical in both languages

Approach	Score
Bi-Self-Seg-300-18M	**43.62**
Bi-Self-Pret-300-2G	42.72
Bi-NED-Seg-100-18M	39.47
Bi-NED-Pret-300-18M	41.85
Bi-NED-Seg-300-18M	39.37

Table 4: Results on the Marathi WordSim task for self-mapping and NED strategies, using different Hindi embeddings. The approach is named in the following format: Bi-<mapping_method>-<hin_embs>-<dims>-<hin_tokens>.

word pairs i.e. 61% of the dataset.[11] Surprisingly, we see a significant dip in the performance of Hin-Pret-300-2G on these word pairs as compared to Mar-Pret-300-60M, indicating that while the word pairs may appear identical in both languages to a native speaker, their usage in the corpora or interaction with other words from the language is different.

7.2 Normalized Edit Distance (NED)

Our NED models use only Hindi embeddings, and project Marathi morphs onto Hindi morphs in the manner mentioned in Algorithm 1. For further simplicity, we also tried a self-mapping; i.e. we do not perform any projection, but calculate the (Hindi) embeddings of the Marathi morphs as they are. Note that this is only possible because Marathi and Hindi share a common script. See Table 4 for the results on different combinations of embeddings and mappings.

Firstly, we observe that the self-mapping performs better than NED in general.[12] This is largely unsurprising; NED would only perform better for Marathi words that are cognates with Hindi words and show a slight difference in spelling; it will perform competitively with self-mapping for identical words in Hindi and Marathi. As we discuss in Section 7.1, such words form a large part of the evaluation dataset. As for the remaining words, it seems that the Hindi embeddings are able to capture the meaning of the unknown Marathi morphs, perhaps due to similarities at a subword level. Applying the NED mapping, however, can result in arbitrary Hindi words with possibly different subwords that may share no semantics with the original Marathi word.

Approach	Comp.	Score
(Mar-Base-300-0.8M	-	24.64)
Bi-Iter-Pret-300-0.8M-2G	Sum	44.28
Bi-Iter-Seg-300-0.8M-9M	Sum	49.49
Bi-Iter-Seg-300-0.8M-18M	Sum	49.21
Bi-Iter-Seg-300-0.8M-18M	First morph	50.06
Bi-Iter-Seg-300-0.8M-36M	First morph	**50.10**

Table 5: Iterative approach results on Word Sim task using different sets of Hindi embeddings for the crosslingual transfer. Format of the approach name: Bi-Iter-<hin_embs>-<dims>-<mar_tokens>-<hin_tokens>. **Comp.**: Composition function.

Another interesting observation is that the Bi-Self-Seg-300-0.8M-18M performs a little better than Bi-Self-Pret-300-0.8M-2G. This affirms our intuition in Section 5 that segmentation on the Hindi side may indeed "visibilize" the subwords that Hindi and Marathi have in common, leading to better performance of the Hindi embeddings on a Marathi evaluation set.

7.3 Iterative Approach

This approach trains Marathi bilingual embeddings from Hindi and Marathi monolingual embeddings. The initial Marathi embeddings used are always the monolingual FastText Mar-Seg-300-0.8M, whereas we try with some different Hindi embeddings. See results in Table 5.

There are three points of interest in the results:

1. We see that the Bi-Iter-Seg-300-0.8M-18M outperforms Bi-Iter-Pret-300-0.8M-2G; i.e. once again, we find that it is better to use embeddings trained on segmented Hindi data for the transfer, even though Hin-Seg-300-18M is trained on two orders of magnitude

[11] Many of these are transliterations of English words. 24 of the total 135 unique words in the dataset are transliterations, and they occur 40 times i.e. 19.6% times in the 104 word pairs.

[12] Note that there is a difference between the self-mapping model and directly applying Hin-Pret-300 as in Table 2 In the former, we segment the Marathi word ourselves and apply Hindi embeddings to the resulting subwords; in the latter, we leave it up to FastText. We note that the former does better.

fewer data than Hin-Pret-300-2G. Since this approach is explicitly bilingual and attempts to project the Marathi and Hindi embeddings into a shared space, this is a much more direct affirmation that the similarities between Hindi and Marathi are best exploited at the subword level from *both* sides.

2. We see that the "first-morph" manner of composition does slightly better than summing or averaging[13] the subword embeddings.[14]

3. Finally, we see that doubling the amount of Hindi data used to train the initial Hindi embeddings does not help. This indicates that the Hindi data is only useful up to a point; we discuss this further in Section 9.1.

8 Results: Wordnet-Based Synonymy Tests

Due to the small size of the WordSim dataset, we also record a set of results on the WBST, which we generate ourselves as mentioned in Section 4.3.2, to test the consistency of the relative performances of some of the above models. See Table 6 for the scores;[15] scores are calculated as explained in Section 4.3.2.

These results confirm some of the findings from the WordSim results; here are some observations from Table 6.

1. Segmentation helps: Mar-Seg-300-0.8M consistently outperforms the Mar-Base-300-0.8M.
2. The iterative method is the best among the low-resource embeddings.
3. There is little or no difference between Bi-Iter-Seg-300-0.8M-18M and Bi-Iter-Seg-300-0.8M-36M: doubling the Hindi data for the bilingual approach seems not to have much effect on the resulting embeddings.
4. The Mar-Pret-300-60M still performs the best, with a seemingly larger margin than in the WordSim task.
5. As MIN increases, the performance of the low-resource methods generally increases. This is natural; the embeddings perform better on words they have seen more frequently in the corpus.

9 Discussion

Some of the clearer findings of our experiments are as as regards segmentation and the benefits of a non-trivial bilingual embeddings transfer.

We see repeatedly that segmentation on both sides of the transfer helps the quality of the LRL embeddings. Segmenting the Marathi data causes a large boost in monolingual performance (Table 1); furthermore, when transferring from Hindi embeddings, Bi-Iter-Seg-300-0.8M-18M outperforms Bi-Iter-Pret-300-0.8M-2G (Table 5); the Hindi embeddings used in the latter are trained on 2 orders of magnitude higher (unsegmented) data.[16] This suggests that the interaction between the two languages is indeed facilitated at a subword level, validating our bilingual native speaker intuition about the same. We also see that the iterative approach consistently outperforms both monolingual models Mar-Base-300-0.8M and Mar-Seg-300-0.8M, indicating that bilingual interaction between the related languages is indeed beneficial. This is a good sign for the project of building NLP tools for truly low-resource languages, although the impact of different typologies on this bilingual effect needs to be explored.

Finally, we find that, in agreement with the findings of the papers that investigate subword composition functions (Zhu et al., 2019a,b), the best-performing composition function for subword embeddings seems to be highly task and data dependent; counter-intuitively, even discarding everything except the first subword seems to work better in some cases than aggregating the embeddings of all parts of the token.

9.1 Using Hindi data

To the best of our knowledge, this is the first work that clearly demonstrates that a trivial "copy-and-paste" transfer approach, such as our NED models, is not adequate, even when working with two culturally related languages that share a very high percentage of cognates as well as syntactic and mor-

[13] We do not report averaging scores since they are almost identical to the summing scores

[14] This could be for several reasons; for example, if the first subword approximates the root of the word, then it may capture most of the meaning, whereas the remaining information may be irrelevant or add noise.

[15] Note that since a synonym as well as the detractors are selected randomly from the Wordnet, the scores show some variation over different runs; however, these scores are representative of the general trend of performance.

[16] Of course, we are talking about performance in terms of the resultant Marathi bilingual embeddings rather than the direct evaluation of the Hindi embeddings.

(MIN, N)	Test size	Mar-Base -300-0.8M	Mar-Seg -300-0.8M	Bi-Iter-Seg-300-0.8M-18M	Bi-Iter-Seg -300-0.8M-36M	Mar-Pret -300-60M
(10,6)	1183	51.23	58.92	**61.62**	57.06	**84.70**
(10,5)	1183	51.90	54.78	58.66	**61.54**	**84.87**
(20,6)	684	48.98	53.65	**59.94**	58.19	**84.50**
(20,5)	684	57.89	59.94	**64.47**	64.33	**87.57**
(50,5)	293	58.02	63.14	67.24	**68.94**	**81.23**

Table 6: WBST Results. MIN refers to the minimum frequency of the question and options in the corpus, N refers to the number of total options including the answer. Summing was used as the composition function where applicable. The two best-performing models have been bolded.

phological properties. Our experiments with identical words pairs in Table 3 especially show that even identical words that are not false friends may behave differently depending on the language;[17] using Hindi embeddings *directly*, even for identical words, is problematic.[18] We believe that this is an important insight into embeddings transfer that rejects relying on trivial or simplistic approaches.

Many of our experiments are intended to indicate how useful the Hindi data and embeddings are to the Marathi tasks; e.g. we evaluate Hin-Pret-300-2G directly on the Marathi WordSim task (Table 2), we experiment with different amounts of Hindi data for both tasks (Tables 5 and 6), and we try a self-mapping with the NED model (see Table 4). We see that doubling or halving the amount of Hindi data does not boost the results for either task and sometimes even harms performance. Similarly, we see that Joint-Seg-300-0.8M-18M performs worse than Mar-Seg-300-0.8M (see Table 1). In conjunction, these results imply that under the current transfer paradigm, adding more Hindi data may sometimes hurt rather than benefit; too much Hindi data for the purpose of training bilingual embeddings may actually “conceal” Marathi word interactions. We invite further investigation of this effect.

10 Future Work

This work is intended to be the pilot in a series of similar studies. We hypothesize that we can obtain similar results for other genealogically related LRL-HRL pairs. We intend to repeat these experiments for language pairs[19] such as Punjabi-Hindi, Assamese-Bengali, Konkani-Marathi, and others. Some of the issues we will be working against are different scripts, morphemic segmentation of unknown languages, and the lack of evaluation data. We would also like to experiment with the integration of parallel data into this approach.[20] Finally, we also think it would be interesting to extend this problem from a bilingual to a multilingual one, with multiple sources for a target language. This would be highly pertinent in the case of Indian languages, where even major Indian languages may be interconnected, and regional languages may benefit from the resources of more than one HRL.

11 Conclusion

Embeddings transfer from a high-resource to a low-resource language with the incumbent data asymmetry and nearly no parallel data is an important task in today’s scenario of data inequality across languages. We target a family of geographically and genealogically related languages, including some high-resource languages and other low-resource languages, possibly undergoing digitization and data collection. In specific, we look at the Indian languages, and assume that both languages are morphologically rich. We take such a language pair i.e. Hindi-Marathi, and artificially simulate a low-resource scenario for Marathi. We present an approach to embeddings transfer that uses very little monolingual data on the LRL side, and no parallel data, and we demonstrate that it improves significantly over a monolingual FastText baseline for both the WordSim and WBST tasks. Further, its performance on the former task is close to that of high-resource pretrained FastText embeddings.

[17] This is to say even if words a and b occur identically and with the same senses in both languages, the word pair (a, b) may have a different relationship depending on the language.

[18] Note that Hin-Pret-300-2G performs very well on the Hindi WordSim dataset; its monolingual quality is not the problem.

[19] simulating LRL environments

[20] We mention one way of doing this in Section 6.2.

References

Syed Sarfaraz Akhtar, Arihant Gupta, Avijit Vajpayee, Arjit Srivastava, and Manish Shrivastava. 2017. Word similarity datasets for Indian languages: Annotation and baseline systems. In *Proceedings of the 11th Linguistic Annotation Workshop*, pages 91–94, Valencia, Spain. Association for Computational Linguistics.

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.

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.

Ondřej Bojar, Vojtěch Diatka, Pavel Rychlý, Pavel Straňák, Vít Suchomel, Aleš Tamchyna, and Daniel Zeman. 2014. HindMonoCorp 0.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Aditi Chaudhary, Chunting Zhou, Lori Levin, Graham Neubig, David R. Mortensen, and Jaime G. Carbonell. 2018. Adapting Word Embeddings to New Languages with Morphological and Phonological Subword Representations.

Chakrabarti Debasri, Narayan Dipak Kumar, Pandey Prabhakar, and Bhattacharyya Pushpak. 2002. Experiences in building the Indo-Wordnet: A Wordnet for Hindi. In *Proceedings of the First Global WordNet Conference*.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. *arXiv preprint arXiv:1802.06893*.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: Hlt*, pages 771–779.

Bradley Hauer, Garrett Nicolai, and Grzegorz Kondrak. 2017. Bootstrapping unsupervised bilingual lexicon induction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 619–624.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020. inlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4948–4961.

Efsun Sarioglu Kayi, Vishal Anand, and Smaranda Muresan. 2020. Multiseg: Parallel data and subword information for learning bilingual embeddings in low resource scenarios. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 97–105.

Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition*, pages 9–16.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space.

Ritesh Panjwani, Diptesh Kanojia, and Pushpak Bhattacharyya. 2018. pyiwn: A Python based API to access Indian Language WordNets. In *Proceedings of the 9th Global Wordnet Conference*, pages 378–383.

Maciej Piasecki, Gabriela Czachor, Arkadiusz Janz, Dominik Kaszewski, and Paweł Kędzia. 2018. Wordnet-based evaluation of large distributional models for Polish. In *Proceedings of the 9th Global Wordnet Conference*, pages 229–238, Nanyang Technological University (NTU), Singapore. Global Wordnet Association.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Manish Sinha, Mahesh Reddy, and Pushpak Bhattacharyya. 2006. An approach towards construction and application of multilingual Indo-Wordnet. In *3rd Global Wordnet Conference (GWC 06), Jeju Island, Korea*. Citeseer.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ruochen Xu, Yiming Yang, Naoki Otani, and Yuexin Wu. 2018. Unsupervised cross-lingual transfer of word embedding spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2465–2474.

Yi Zhu, Benjamin Heinzerling, Ivan Vulić, Michael Strube, Roi Reichart, and Anna Korhonen. 2019a. On the importance of subword information for morphological tasks in truly low-resource languages. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 216–226.

Yi Zhu, Ivan Vulić, and Anna Korhonen. 2019b. A systematic study of leveraging subword information for learning word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 912–932.