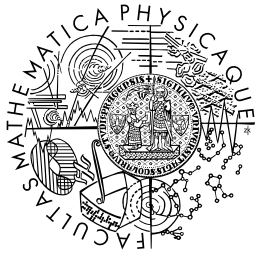

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Niyati Bafna

Empirical Models for an Indic Language Continuum

Institute of Formal and Applied Linguistics, Charles University,
Department of Computational Linguistics and Phonetics, Saarland University

Supervisors of the master thesis: doc. Ing. Zdeněk Žabokrtský, Ph.D.,
Dr. Cristina España-Bonet,
Dr. Josef van Genabith
Study programme: Language Technologies and Computational Linguistics, Charles University,
Language Science and Technology, Saarland University

Prague 2022

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

First and foremost, I would like to thank my advisors for their help and guidance: doc. Ing. Zdeněk Žabokrtský, for his perpetual enthusiasm about the project, large-picture perspective, and readiness to discuss every idea of mine (including the bad ones); Dr. Cristina España-Bonet, for her patience and encouragement, and for the on-point advice in response to all my long emails; and Dr. Josef van Genabith, not only for his contributions to the project but also for helping me out at a low point of time for me in the semester, and giving me extremely good counsel - academic and general. You are inspirations to me both as researchers and teachers, and I feel extremely lucky to have worked with you.

A special thanks to Dr. Markéta Lopatková, and Ms. Bobbye Pernice for seeing me successfully through the LCT rollercoaster, my professors at Charles and Saarland University, and Dr. Samar Husain - I loved learning from all of you.

Thanks to my friends in Prague and Saarbrücken - Michael, Saad, Dota, Allison, Siyu, Alina, Averie, Zahra, Sonal - for lighting up the dark days spent editing LaTeX tables. We had good times, many of them around a hot pot, and I'm not sure how much fun any of this would have been without them. I'd like to thank my parents, who have always seemed to have great faith in me, for never having doubts that "ho jayega" - it'll work out - and for sending me food. And of course, (no) thanks to my siblings, Mitali, Tanvi, and Tanish, for living life side-by-side with me. Finally, Ananya and Barun, thanks for listening to me, for making me laugh at myself, and in advance, for proof-reading.

Title: Empirical Models for an Indic Language Continuum

Author: Niyati Bafna*†

Supervisors: doc. Ing. Zdeněk Žabokrtský*, Ph.D.,
Dr. Cristina España-Bonet†,
Dr. Josef van Genabith†

Institutes: *Faculty of Mathematics and Physics: Institute of Formal and Applied Linguistics, Charles University,
†Department of Computational Linguistics and Phonetics, Saarland University

Abstract:

Many Indic languages and dialects of the so-called "Hindi Belt" and surrounding regions in the Indian subcontinent, spoken by more than 100 million people, are severely under-resourced and under-researched in NLP, individually and as a dialect continuum. We first collect monolingual data for 26 Indic languages and dialects, 16 of which were previously zero-resource, and perform exploratory character, lexical and subword cross-lingual alignment experiments for the first time on this linguistic system. We present a novel method for unsupervised cognate/borrowing identification from monolingual corpora designed for low and extremely low resource scenarios, based on combining noisy semantic signals from joint bilingual spaces with orthographic cues modelling sound change; to the best of our knowledge, this is the first work to do so, especially in a (truly) low-resource setup. We create bilingual evaluation lexicons against Hindi for 20 of the languages, and show that our method outperforms both traditional orthography baselines as well as EM-style learnt edit distance matrices, showing that even noisy bilingual embeddings can act as good guides for this task. We release our crawled data in a new collection called "HinDialect"; we also release the evaluation data, code, and results here:
`https://github.com/niyatibafna/north-indian-dialect-modelling`

Keywords: dialect continuum Indic North Indian cognate induction bilingual embeddings

Contents

Introduction

Hindi is listed as one of the official languages of India. It is often claimed to be the most widely spoken in the country, especially in North India, with the latest census showing 43.63% of Indians that have Hindi as their mother tongue.[1] However, this figure counts speakers of the languages of the Indo-Aryan dialect continuum that stretches from Rajasthan in the West to Bihar and Jharkhand in the East, called the Hindi Belt, or the Hindi-Urdu Belt, of which modern standard Hindi[2] is only a part.

This continuum, spread out over North and Central India, contains a wide variety of languages and dialects that are often mutually unintelligible, and form subgroups of their own. For example, we have the Rajasthani languages, including Marwari, Mewari, Nimaadi, and others; the Bihari languages, such as Bhojpuri, Magahi, or Awadhi; the Pahari languages, like Nepali, Garwali, Kumaoni, and so on.[3] Many of the languages we work with are not used in primary or higher education institutions, and lack support from the state governments where they are spoken. This has also culminated in the perception of local languages as "rough", unsophisticated, and associated with uneducated populations. The official language of the following states: Bihar, Rajasthan, Haryana, Himachal Pradesh, Madhya Pradesh, Uttarakhand, Uttar Pradesh, Delhi, Jharkhand, and Chattisgarh, is Hindi, sometimes in conjunction with some native variants (Chattisgarhi is recognized as a co-official language of Chattisgarh, similarly for Maithili in Bihar); native languages are not given any official status and are usually clubbed together as "Hindi". This has led to protest for language recognition in some states e.g. for Maithili in Bihar, resulting in official recognition and support for Maithili in 2003.[4]

[1] See here for the 2011 census:
`https://en.wikipedia.org/wiki/2011_Census_of_India`.
This is the latest census because the 2021 census was delayed due to the COVID-19 pandemic.

[2] Hindi and Urdu are political variants with mainly lexical differences, with Urdu showing more Persian influence in its vocabulary. The term "Hindi-Urdu" (usually used in linguistics) refers to the language "Hindustani", which is a blend of Hindi and Urdu, used widely as the lingua franca of North India and Pakistan and given official recognition as late as in the 20^{th} century. Around the time of the Partition of India, India and Pakistan chose to recognize Hindi and Urdu separately as (lexically) Sanskritized and Persianized versions of Hindustani respectively, although they are largely mutually intelligible. Note that while Hindustani was written in the Devanagari, Kaithi as well as Perso-Arabic scripts, Urdu is written in the Perso-Arabic script and Hindi in Devanagari. For our purpose, we use the terms Hindi and Hindi-Urdu interchangeably.

[3] See `https://glottolog.org/resource/languoid/id/indo1321` for the full language tree.

[4] See an overview of the Maithili Movement here: `https://frontline.thehindu.com/books/article24200882.ece`.

This political-linguistic situation means that NLP resources for these languages are sorely lacking or non-existent; most of these languages, despite having millions of speakers (e.g. Marwari) have little or no monolingual data, or other basic resources such as lexicons, grammars, taggers, embeddings, etc. Collecting data from the web can be tricky for a few reasons: firstly, digital presence for many of these languages is low, and secondly, we lack good quality language identification tools to automatically differentiate between closely related languages, written in the same script, in crawled text.

This project works with 26 languages, all written (primarily) in Devanagari, with the exception of Sindhi, which is written primarily in the Perso-Arabic or Naskh script, but also in Devanagari. We categorize them as Band 1, 2 or 3, according to how well-resourced they are and how much attention is given to them in NLP, with Band 1 consisting of the best resourced languages, and Band 3 containing previously "zero-resource" languages. By this term, we mean languages for which it is virtually impossible to find standardised NLP datasets. Our main contributions consist in collecting monolingual resources for the languages under consideration, as well as presenting a novel strategy for unsupervised bilingual cognate/borrowing lexicon induction in low-resource scenarios, taking on this problem for the first time with the Indic dialect continuum. Note that while we do have lexical resources for Band 1 and 2 languages including WordNets for some Band 1 languages that can provide some supervision, we simulate low-resource unsupervised settings for these languages consistent with the truly low-resource Band 3 languages, using the WordNets when available only for evaluation.

Note that in this work, we do not distinguish between cognates, which are words in related languages with shared etymology, usually descended from a single ancestor, and borrowings, which are words that have simply been adopted as are from any language regardless of the genealogy of either language; henceforth, we use the term "cognate" as including borrowings. This is because our main concern is to build bilingual lexicons that are as large as possible, regardless of the reason for or type of lexical equivalents contained in them.

First, we gather monolingual data for these languages, forming the largest collection (in the number of languages) of a dialect continuum as far as we know. This also introduces the first ever monolingual data for 16 zero-resource languages to the NLP community - 15 Indic/Indo-Aryan (IA) languages, and Korku. In general, such a corpus has wide applications for work in crosslingual transfer of NLP tools and models, historical linguistics, dialect continua, and building language support for these communities. We probe the resulting multilingual collection at a character, subword and lexical level, finding a general link between relatedness and genealogically and geographically proximal languages.

We use the corpus for cognate induction (CI) for each target language with Hindi. We identify cognates from monolingual corpora containing fully inflected word forms[5] in a completely unsupervised manner. We also assume asymmetric data scarcity; i.e. we have abundant monolingual resources for Hindi, but perhaps only a few thousands or ten thousands of tokens in monolingual data for the target language. These constraints set this task apart from most previous literature in cognate identification [List, 2014, Fourrier et al., 2021, List, 2019, Artetxe et al.,

[5]While most literature assumes lemmatized word lists as input for this task, we do not have lemmatizers for these languages.

2018]; however, they are highly realistic when attempting to build resources for truly low-resource languages.

We present two strategies for cognate identification, evaluating on synthetically created test sets. In the first approach, we experiment with iteratively learning substitution probabilities within an edit distance paradigm. Our second approach combines noisy semantic signals from a subword embedding space with orthographic distance measures. reporting qualitative improvements over the baseline. As a byproduct of the work, we also make available to the community the first collection of bilingual embeddings for 16 languages (with Hindi), as well as evaluation data for 20 languages.

In summary, our main contributions for research for the Indic dialect system are the following: we contribute **data** in the form of monolingual corpora for 26 languages and evaluation lexicons for cognate induction for 20 languages. We present **strategies** for cognate induction adapted to an asymmetric low resource scenario, potentially useful for other NLP tasks. This work may also be relevant to research in other low resource dialect continua around the world[6] for which we not as yet have bilingual resources. Finally, we make **bilingual embeddings** available for the languages under consideration. We hope that this study kickstarts research into the Indic dialect continuum, potentially providing language support to hundreds of millions of speakers.

We release our collected monolingual data in the form of a new dataset called "HinDialect" here:
`http://hdl.handle.net/11234/1-4787`.
All our code and results are available here:
`https://github.com/niyatibafna/north-indian-dialect-modelling`.

[6]See `https://en.wikipedia.org/wiki/Dialect_continuum` for more examples of dialect continuua, such as the Turkic and Arabic languages.

Chapter 1

The Indic language continuum

See a map of the Hindi Belt languages in Figure 1.1.[1] See Table 1.1 for a list of the languages we are working with, including information about where they are spoken, and they phylogenetic subcategorization in the Indo-European family. While we collect data for a few other languages as well and include them in our exploratory experiments on the continuum, this is the set of languages for which we perform cognate induction.

There are 26 languages listed in total. These languages were chosen according to availability in our data source (discussed in Chapter 3. Further, we filtered languages by script; those written in Devanagari were given preference; we chose to exclude languages such as Gujarati or Bengali. Given this consideration, Sindhi provides an interesting case since it is written both in Devanagari and in the Perso-Arabic script depending on where it is being used, with most of its NLP resources in the latter; however, we finally chose to include this language.

The language Korku is also an outlier: Korku is the western-most member of the Austro-Asiatic Munda group of languages in India, and the only non-Indic language that we consider. We only retain it as a sanity check in our experiments; we expect this language to do badly on many similarity metrics, against which we can contrast the performance of genealogically related languages. Note that Korku is different from Koraku (also written as "Kodaku"), which is a term for a related Austro-Asiatic Munda group of languages. We have no data for and do not work with Koraku; in the following sections and figures, we are always referring to Korku.

1.1 Notes on terminology

1.1.1 Umbrella languages

Some of the languages listed in Table 1.1 we are working with can be considered a group of different dialects; e.g. Rajasthani is a term used to refer to a group of languages spoken in Rajasthan, including Marwari, Mewari, and others. Similarly, Bhili and Chattisgarhi are also groups of dialects/languages within themselves. We use these terms as referring to a single language due to lack of a finer-grained classification in available data.

[1] This map contains most of the languages we are working with but not all; e.g. Angika.

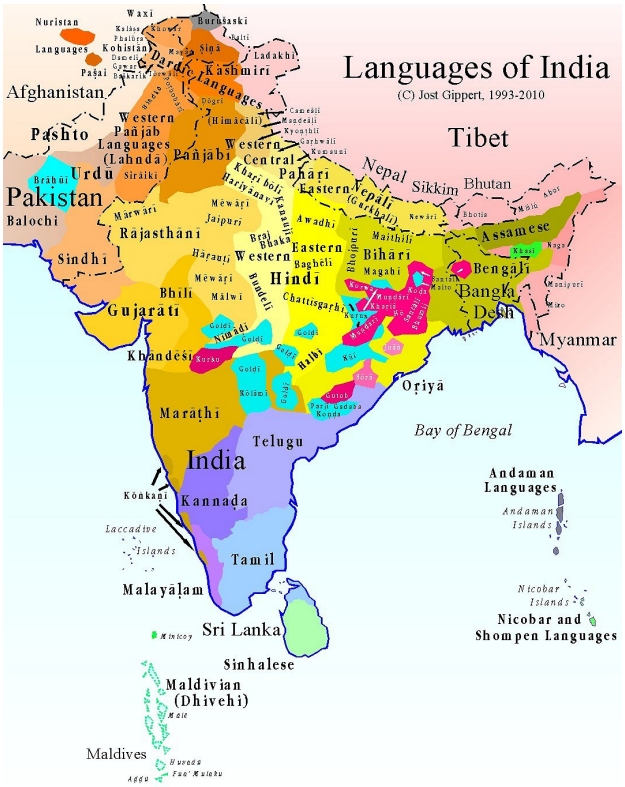

Figure 1.1: Showing the different languages of the Hindi Belt. Taken from `https://titus.fkidg1.uni-frankfurt.de/indexe.htm`

1.1.2 "Hindi" and related terms

The term "Hindi" has a narrow sense as in "modern standard Hindi", which is largely a political and educational construct, and a broad sense which encompasses many other languages and dialects such as Brajbhasha, Haryanvi, Bhojpuri, and others, as in the "Hindi" belt. Sometimes, eastern languages such as Bhojpuri and Magahi are excluded from this umbrella, in which case "Hindi" refers to the phylogenetic sub-family of Western Hindi languages. The term "Hindustani" is equivalent to "Hindi-Urdu", referring to the blend of Hindi and Urdu resulting from a 18th century fusion of Khadi Boli (an ancestor of Hindi), and the Persianized Awadhi during the Mughal rule. Finally, "Khadi Boli" can be considered the prestige dialect of Hindi, and is closest to what we think of as modern standard Hindi. However, the term "Hindi" is obviously sociopolitically and linguistically distinct from "Khadi Boli"; we will not use the two interchangeably. In the following sections, we will do our best to make our meaning as clear as possible when using the above terms.

1.1.3 The "Indic dialect continuum", or the "Hindi Belt"

The term "Indic" may be used both to refer to Indo-Aryan languages, as well as all languages spoken in the Indian subcontinent regardless of their genetics. In the work, we always apply the former sense of this term. We use the term "Indic dialect continuum" to refer to the Indo-Aryan continuum across North

India and parts of Nepal and Pakistan. The part of this continuum that has to do with Hindi and so-called dialects of Hindi is sometimes referred to as the "Hindi Belt", although the latter term may also have political implications. Most of the languages we work with are generally considered part of the Hindi Belt, but not all (e.g. Nepali). There are also several related Indic languages that we do not work with, such as Gujarati, Assamese, and Odiya; these languages are written primarily in different scripts, and are more distant relatives to Hindi than languages such as Bhojpuri or Rajasthani, which are the focus of this work. In summary, either term "Indic dialect continuum" and "Hindi Belt" do not exactly describe our set of languages, and we use both only as approximations.

1.2 Classification

Since the languages we are dealing with cover a range of resource situations, we divide them into three categories, based not only on the amount of resources they have but also the amount of attention given to them in NLP research.

- **Band 1**: This contains the best resourced languages, i.e. Hindi, Nepali, Sindhi, and Marathi. These languages have pre-existing corpora containing gigatokens of monolingual data, pretrained embeddings, tokenizers, and basic evaluation resources. They also usually have focused NLP research concerning improvement of these resources. However, note that some of these languages are often considered mid-to-low resource languages on an absolute or global scale, for the reason that they often lack labelled corpora, crosslingual data, and other more advanced resources that we have for some European languages. [Joshi et al., 2020]

- **Band 2**: This contains languages in our list for which we have monolingual data ranging around tens of thousands of sentences, and some annotated corpora for basic tasks, namely, Bhojpuri, Magahi, Awadhi, Maithili, and Braj. While there is some research for these languages, it is still in a fetal stage, generally focusing on developing basic resources and/or describing the relationship of these languages to Hindi or other more prominent languages.

- **Band 3**: This comprises the other 16 languages, such as Himachali, Baiga, and Nimaadi, for which we have no systematic resources available to the NLP community. By this, we mean to exclude one-off content such as bloggers, native speakers providing manual translations of words from their language, etc. Such information is difficult to organize and collate, besides concerns about legitimacy. Note that while Table 1.1 lists 17 Band 3 languages, we only talk about 16 Band 3 languages in the rest of this work. This is because the status of one of these langauges: Khadi Boli, is ambiguous in this regard. Khadi Boli is considered very close to modern Hindi, for which, of course, we have abundant data. While we maintain the distinction between Hindi and Khadi Boli in our experiments, we cannot claim to have collected the first monolingual data for this variety; therefore, we only talk about 16 Band 3 previously zero-resource languages.

Language	Primary Regions	Language (Sub-)Family	Data (Tok.)	Native sp.
BAND 1				
Hindi	Uttar Pradesh*, Bihar*, Rajasthan*, 13 others	IA Central, Western Hindi	1.86B[1]	250M†
Marathi	Maharastra*, Goa*	IA Southern, Marathic	551M[1]	73M
Sindhi	Sindh*, Pakistan, Rajasthan Gujarat,	IA Northwestern, Sindhi-Lahnda	61M[5]	25M
Nepali	Nepal*, West Bengal*	IA Northern, Eastern Pahari	14M[2]	16M
BAND 2				
Bhojpuri	Bihar, Jharkhand*	IA, Bihari	259K[3]	40M
Magahi	Bihar, Jharkand*	IA, Bihari	234K[3]	40M
Awadhi	Bihar	IA, Bihari	123K[3]	38M
Maithili	Bihar*, Jharkhand*	IA, Bihari	300K[4]	14M
Brajbhasha	Uttar Pradesh	IA Central, Western Hindi	249K[3]	1M
BAND 3				
Rajasthani	Rajasthan	IA Central, Gujarati-Rajasthani	-	50M
Chattisgarhi	Chattisgarh*	IA Central, Eastern Hindi	-	18M
Angika	Bihar, Jharkhand*	IA, Bihari	-	15M
Hariyanvi	Haryana, Rajashtan	IA Central, Western Hindi	-	13M
Bajjika	Bihar	IA, Bihari	-	12M
Kannauji	Uttar Pradesh	IA Central, Western Hindi	-	9.5M
Garwali	Uttarakhand	IA Northern, Central Pahari	-	6M
Bundeli	Madhya Pradesh, Uttar Pradesh	IA Central, Western Hindi	-	5.6M
Malwi	Rajasthan, Madhya Pradesh	IA Central, Bhil	-	5M
Bhili	Rajasthan, Gujarati, Madhya Pradesh	IA Central, Bhil	-	3M
Himachali	Himachal Pradesh	IA Northern, Himachali	-	2M
Kumaoni	Uttarakhand	IA Northern, Central Pahari	-	2M
Nimaadi	Rajasthan, Madhya Pradesh	IA Central, Bhil	-	2M
Korku	Madhya Pradesh, Maharashtra	Austro-Asiatic, North Munda	-	0.7M
Bhadavari	Jammu Kashmir	IA Northern, Western Pahari	-	0.1M
Baiga	Chattisgarh	IA Central, Chattisgarhi	-	UNK
Khadi Boli	Delhi	IA Central, Western Hindi	-	UNK

Table 1.1: Language bands. "Regions spoken" only mentions places in the Indian subcontinent; * indicates official status. Speaker counts taken from (latest) 2011 census. [1][Kakwani et al., 2020], [2][Yadava et al., 2008], [3][Zampieri et al., 2018], [4][Goldhahn et al., 2012] [5][Conneau et al., 2019]. †: probably inflated

Chapter 2

Background and Related Work

We describe related work in different areas of NLP and linguistics that are relevant to our project.

2.1 Data and Resources

As we mentioned, Band 1 languages have relatively high amounts of data and resources. AI4Bharat [Kunchukuttan et al., 2020] is one of the largest projects in this regard, that releases monolingual corpora and embeddings for 14 prominent Indian languages. We also have other corpora for individual languages, of course, e.g. HindMonoCorp [Bojar et al., 2014] for Hindi. Further, we have WordNets for Hindi, Marathi, and Nepali, included in the IndoWordNet effort for 18 Indian languages[1] [Sinha et al., 2006, Debasri et al., 2002]. While we list Sindhi as a Band 1 language, note that existing Sindhi resources may be in the Naskh (Perso-Arabic) script; it is also not included in the above IndoWordNet project.

For Band 2 languages, we have some collection efforts, mostly crosslingual but including some parallel data. Zampieri et al. [2018] presented a shared task for language identification for Awadhi, Braj, Bhojpuri, Magahi, and Hindi with 15k sentences for each language; Ojha [2019] presents monolingual 45k monolingual sentences in Bhojpuri as well as English-Bhojpuri parallel data. We have similar works for Magahi [Ojha et al., 2020]. We have 10k sentences in Maithili crawled by Goldhahn et al. [2012] as part of a mass collection effort, as well as a corpus collected by the Linguistic Data Consortium of Indian languages[2], unfortunately not freely available; we also have a translation lexicon to English given by TDIL-DC[3] containing technical internet terms. Mundotiya et al. [2021] collect monolingual corpora[4] for Bhojpuri, Magahi, and Maithili, as well as POS-tagged annotated corpora and WordNets aligned with the larger IndoWordNet effort, presenting baseline tagging accuracies and analyses of crosslingual similarity of these languages compared to Hindi; Mundotiya et al. [2020] presents NER-annotated corpora and trained NER models for the same 3 languages.

[1] See `http://www.cfilt.iitb.ac.in/WordNet/webmwn/`

[2] `https://www.ldcil.org/resourcesTextCorp.aspx`

[3] `https://tdil-dc.in/index.php?option=com_download&task=fsearch&lang=en`

[4] These resources are not publicly available yet.

2.2 Dialect continua

The evolution of and cross-lingual interactions in dialect continua have of course been the object of study of many works of linguistics [Williamson, 2000, Jeszenszky and Weibel, 2015, Heeringa and Nerbonne, 2001]. There are two primary methods of structuring such languages: a phylogenetic tree mode, which is generally the traditional perspective through which language families are viewed, and a wave model, which argues that that the importance of vertical inheritance is inflated, and that it is more principled to model language change using horizontal transfer between proximal languages or dialects [François, 2015]. Much work in computational phylogenetics subscribes to either or both of these models; e.g. Rama and Singh [2009] (arranging Indian languages in phylogenetic trees), and Yamauchi and Murawaki [2016] (contrasting the tree and wave models of flow of typological features).

There is a rich literature in linguistics research, mostly evolution-based, of the Indo-European language family [Egorova and Egorov, 2019, Garrett et al., 2018, Garre, 2006] as well as the Indo-Aryan branch [Li et al., 2018, Allassonnière-Tang and Dunn, 2020]. Recent methods also apply NLP tools for dialectology; Cathcart [2019] and Cathcart and Rama [2020] work with deep neural models of dialectology and wordform prediction respectively for over 50 languages of the Indo-Aryan branch, including 4 Band 3 languages.[5]

Some works also focus on synchronic grammatical or phonological aspects of this continuum, e.g. Phillips [2012] work on grammatical aspects of the Bhil tribal continuum, Mishra and Bali [2011] work on vowel inventories and phonological comparative study of 7 Hindi belt dialects, and Mishra and Bali [2010] work on describing phonological transfer rules from Hindi to some of these languages. Kumar et al. [2018] train language identification for Bhojpuri, Magahi, Awadhi, Braj, and Hindi (Band 2 except Hindi); the VarDial 2018 task on the same also prompted many works for this task and language set. However, in general, there is a paucity in NLP research in Band 2 languages, from the perspective of developing corpora or tools for them. Such research is non-existent for Band 3 languages.

2.3 Contextual LMs for Indian languages

The advent of large transformer-based pre-trained language models i.e. BERT and multilingual BERT [Devlin et al., 2018] opened many possibilities for low-resource languages to ride on data in other languages [Pires et al., 2019, Wang et al., 2020] Kakwani et al. [2020] presented "Indic BERT" pre-trained on 11 Indian languages, which have similar orders of magnitude as our Band 1 languages. Dhamecha et al. [2021] investigate how BERT-like models that have been pre-trained on a particular subset of Indo-Aryan languages respond to finetuning on another related language, showing that low-resource languages like Punjabi benefit most, and for each language, the most beneficial pre-trained subset differs. Note that Punjabi is still a Band 1 language; however, relative to European languages, many Indic Band 1 languages are considered low-resource languages.

Most research in large multilingual LMs is in Band 1 languages since these lan-

[5]The data used is unfortunately not available.

guages have the data and (for some languages) the evaluation resources to support this research. While it is feasible that Band 2 languages will soon gain attention in this context given the recent development of corpora for these languages, Band 3 languages are entirely out of the picture even for zero-shot approaches, as far as we know, due to a combination of factors, including low digital presence and lack of any organized data in these languages.

2.4 Multilingual Lexicon Induction

2.4.1 Non-neural methods

Much previous work has been based in *non-neural methods*. Batsuren et al. [2019] use semantic relationships from the Universal Knowledge Core [Giunchiglia et al., 2018] which is built from existing WordNets,[6] gold annotations as well as geographical-orthographic similarity measures for cognate identification. List [2012] induces cognate sets over aligned word lists of languages in a language family by an iterative approach that learns phonological rules from current cognate set, which is implemented in the software LingPy [List, 2014]. Hall and Klein [2010] works with unaligned word lists for languages in the same family, modelling transfer within a tree-based framework and learning edit-distance based transformation matrices for each vertical edge (representing inheritance) via a similar iterative approach; char-gram models are used to smooth word form predictions from this model. Nicolai et al. [2018] introduce a new system using a character-level transducer to perform several tasks such as cognate identification and morphological inflection; Çöltekin [2019] compares linear and neural models to predict the next edit-distance based transducer action to the related task of crosslingual morphological inflection. Kanojia et al. [2019] identify cognate sets for (Band 1) Indian languages using the IndoWordNet combined with lexical similarity measures, training neural models over the resulting data. In earlier works, Scherrer and Sagot [2014], inspired by the seminal work by Koehn and Knight [2002], induced cognates sets in a completely unsupervised manner using orthographic similarities leveraged by a character-based alignment algorithm, as well as context vectors based on co-occurrence counts in monolingual corpora. Although the idea of learning edit distance matrices is quite old [Bilenko and Mooney, 2003], it has not been used in combination with modern embeddings-based methods for cognate identification as far as we know.

2.4.2 Neural and embeddings-based methods

Conneau et al. [2017] was one of the earliest works to link bilingual lexicon induction with bilingual embedding spaces, or the alignment of monolingual embeddings. This idea has since then been explored by other works that seek to adapt it to low-resource settings or relax its strong isometry assumption [Dou et al., 2018, Patra et al., 2019],[7] sometimes using a bootstrapping strategy for

[6] CogNet contains only Band 1 Indic languages.

[7] Isometry is a property of two graphs by which they have the "same shape", which is to say that they can be rotated to map onto each other perfectly. With respect to embedding spaces, it is used more as an approximation to describe different language spaces that have equivalent

embeddings alignment and bilingual lexicon induction [Artetxe et al., 2018, Cao and Zhao, 2021]. More recently, we also have works using contextual embeddings for bilingual lexicon induction (BLI); Schuster et al. [2019] averages the contextual embeddings of a word over different context to form anchor vectors for words, Zhang et al. [2021] take this idea further to combine static and contextual vectors for BLI. Fourrier et al. [2021] frame cognate detection as a machine translation problem, finding that SMT still beats NMT over smaller datasets; Kanojia et al. [2019] identify cognate sets for (Band 1) Indian languages using the IndoWordNet combined with lexical similarity measures, training neural models over the resulting data.

2.5 Other related work

Hedderich et al. [2021] survey strategies for low-resource NLP, such as data augmentation, transfer learning, and others, in the context of machine learning and deep learning. Joshi et al. [2020] provide a taxonomy of languages based on their data availability and survey the inclusivity of NLP conferences to different languages types.

Many works dealing with low-resource dialects or variants focus on neural or statistical machine translation. We have several such studies for the Arabic language continuum; these works focus on creating parallel corpora [Meftouh et al., 2015], solving script problems [Guellil et al., 2017], and multitask and unsupervised setups for shared representations and transfer [Baniata et al., 2018, Farhan et al., 2020]. Lakew et al. [2018] experiment with translation from English into two varieties each of Portuguese and French given data in each variety, either labelled for variety or not. Wan et al. [2020] train neural dialect translation from Mandarin to Cantonese using the concept of pivot (shared) and private (dialect-specific) dimensions of word embeddings. There is some research in this genre for Band 1 and 2 Indian languages; Madaan and Sadat [2020] attempt data augmentation for MT from Sindhi, Bhojpuri, and Magahi to English by, for example, switching source and target side. Kumar et al. [2020] present zero-shot approaches to NMT between Hindi, Bhojpuri, and Magahi as part of the LoResMT 2020 shared task Ojha et al. [2020].

Research in subword-based neural strategies are also relevant to us since we are dealing with data-scarce low-resource languages; Faruqui et al. [2016] perform morphological inflection using a char-level encoder-decoder model with grammatical features as well as the lemma as input. Jha et al. [2018] perform incorporate word transduction from Hindi to Bhojpuri as an attempt to deal with OOV words in NMT. Ataman et al. [2019] present a strategy to deal with OOV words in morphologically rich languages by character-level decoding with explicit architecture for inflection-handling.

words roughly the same distance from each other.

Chapter 3

Data collection

The objective of this stage is to collect a multilingual corpus with data labelled for the dialect/language that it is in. We would like to have enough languages to reasonably represent the Indic continuum.

3.1 Is there anything out there?

Searching the web for content in individual Band 3 languages yielded some blogs and YouTube song videos (with presumably same-language associated comments) for a handful of languages, and nothing for most of them. For example, for Rajasthani, we have `https://www.maruwani.com`, containing 18 articles in Rajasthani,[1] as well as a Rajasthani-English bilingual dictionary.[2] In general, we face the following problems:

- Such material seems to be scantily available over our 16 languages of interest. It's possible that focused searches will yield better results; however, automating that for all languages (for example, with some kind of a bootstrapping strategy) in a manner that preserves quality of results would be a difficult task in itself.
- Automated crawling from website such as YouTube or personal blogs will require sophisticated language identification, especially since all these languages are written in the same script (i.e. Devanagari). Scraping a good quality monolingual corpus for any language would therefore entail distinguishing between closely related dialects; of course, building such a tool would require data in the first place.

3.2 Kavita Kosh

Kavita Kosh: `http://kavitakosh.org/kk/`, translating roughly to "poetry collection", is an online collection of folksongs and poems in 31 languages from

[1] As we have mentioned, Rajasthani is a collection of languages, traditionally associated with Marwari and Mewari, but including the Bhil tribal languages and several others spoken in nearby states. We assume that the author means "Marwari" going by one of their posts.

[2] Interestingly, it also contains an English post:
`https://www.maruwani.com/2009/11/are-you-rajasthani.html` about the employment-related ill-effects of Hindi hegemony over local languages such as Marwari in Rajasthan.

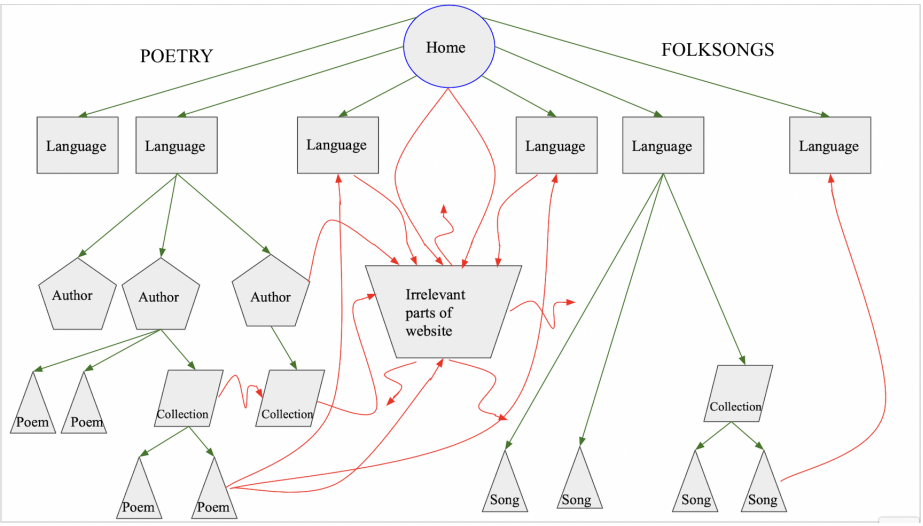

Figure 3.1: Layout of Kavita Kosh website. Green arrows represent links we would like to follow, that adhere to an easily exploitable structure, whereas red arrows represents links that form cross or back edges, interfering with the above structure, or lead to irrelevant content. Links leading out of irrelevant parts of the website may lead to any or all webpages that we are interested in.

on the IA continuum, from Sindhi and Rajasthani in the West, to Marathi in the South, Kashmiri and Himachali in the North, and Bengali in the East. Content is manually curated; the folksongs were collected by the organization, whereas the poetry consists of works by early contemporary writers, mostly from the late twentieth century.

3.3 Format of the website

- **Script**: The Kavita Kosh website is written entirely in the Devanagari script, and all non-content material is in Hindi. Most of the languages that it has poetry for are also written in Devanagari; however, when that is not the case (such as with Gujarati or Bangla), the poems may be written either in transliterated Devanagari or in the original script. Some pages have the option of toggling between different scripts; however, in general, the default is Devanagari.

- **Layout and navigation**: Content on Kavita Kosh is separated into two primary categories i.e. folksongs and poetry, and further segregated by language and author. The basic layout of the website is as shown in Figure 3.1. For folksongs, some pages may have further thematic or other classifications; for example, the Angika folksongs have a preliminary classification by the occasion it is associated with or sung at,[3] and each such occasion may have a page of its own, listing the relevant songs. For poetry, we may also have deeper structure, for example, an author may have written an anthology as

[3]E.g. there are more than 15 genres of wedding songs, to be sung at every stage of the wedding

opposed to individual poems, in which case the website will have a separate page for poems in that anthology.

- **Naming conventions**: Folksong pages are often titled in the style <title_of_folksong>/<language>; however, there are exceptions, and sometimes the language suffix may be different or absent.

  Poetry pages are usually titled <title_of_poem>/<author>, although, again, there are exceptions when the author is unknown, or if the poem belongs to a named collection, or is one of several parts (where it may be suffixed by a simple numeral index). The leaf webpages i.e. containing the poetry/folksongs do not contain any other information pertaining to the language of the content.

- **Other information**: For poetry, author webpages listing all associated poetry may have basic information about the author e.g. date of birth and (if applicable) death and place of origin. This is left blank for many authors, of course, for whom the relevant information may not be known. For folksongs, we do not have dates or exact places of origin.

3.4 Crawling the website

We implement a standard BFS-based crawler with certain modifications to suit our purpose. Crawling the website entails two basic tasks. Firstly, we must retrieve all poetry/folksong content from the website; i.e. we need to ensure we collect all literary content but avoid other pages such as "About" or other functional pages, empty pages with some explanation given,[4] or other parts of the website, such as author descriptions, discussions of some of the poetry, etc. We address this issue by manually pre-specifying constraints over the links to be explored, as well as the some string matches to exclude recurring Hindi material. This, along with the conservative specification of HTML tags used for poetry content, is enough to keep unwanted material from leaking into the dataset.

Secondly, we must correctly associate each piece with its language; this is only non-trivial because as mentioned, the leaf links with the actual content do not contain language tags or language information anywhere in their HTML description. In order to know which language a poem is in, it is necessary to traverse the path from the central page of the website down to the particular poem page and note its ancestor at the level of the tree which groups all content language-wise. Note that, since the poetry, language, and author nodes are also interlinked with each other, there will be several paths to a given leaf node along almost any language-subtree; however, we observe that the *shortest* path from the root to the leaf should have the correct language node ancestor. This is simply a heuristic given the generally uniform structure of the website and the nature of cross-links (authors to other authors, links heading back to the root, to other languages, etc.), meaning that it is quicker to traverse directly to a given leaf via its correct language node ancestor rather than first to another leaf/intermediate

[4]This is because everything other than the folksongs and poems on the website is in Hindi, meaning that if we run into an empty link while crawling for another language and collect the "missing poem" message, we will be introducing Hindi impurities in the dataset.

node, which would link to an ancestor intermediate node (via a cross edge), and then down that path to the target leaf. This is visible from Figure 3.1.

Therefore, we need to account for the following things while crawling:

- We need to have filters for both daughter links as well as collected content.
- We should know at any given state of the BFS which language node ancestor A_L it is descended from. We use a pre-compiled list of languages as written on the website to simplify the recognition of a new language subtree. Given this, we simply link any collected leaf content with A_L.
- We need to have pre-identified "danger zones", in the form of a list of links that the BFS should not explore. This could be because they do not contain pieces (and have material in Hindi), and would lead to time-consuming and non-fruitful exploration of irrelevant parts of the website. It could also be because they ruin the nice structure of the website by providing shortcut paths to other content. Discussion pages may behave in this manner, referring and linking to content or authors from several languages, potentially wreaking havoc with our content-to-language association.
- Since collection is time-consuming (taking about 2 days in total), we also save the BFS state, including all internal variables regularly while crawling; this ensures that in case of interruption, the crawling can resume from where it left off.

Our crawlers are available here:
`https://github.com/niyatibafna/north-indian-dialect-modelling/tree/main/crawlers`

3.5 Crawled data

We store each collected piece in a separate JSON file, with a layout as shown in Figure 3.2. The files are stored in a directory format that is a flatter version of Figure 3.1 for ease of navigation, segregating all files first by genre (poetry/folksongs), then by language. Each *poetry/<lang>* directory, therefore, contains all poetry files in that language; similarly for folksongs. This format throws away groupings by author and publishing information; we considered this as reasonable in order to have a simple a structure as possible. We collect folksongs for 26 languages, and poetry for 18 languages; the number of distinct languages is 31.[5] Poem and token counts are reported in Table 3.1

We collect an average of roughly 100K tokens for Band 3 languages; however, we see wide variation over these languages, with Angika having 1M tokens, and Himachali, Kannaji, and Bhadavari with less than 1K tokens.

We are authorized to release only the folksongs data from this collection: this is available at
`http://hdl.handle.net/11234/1-4787`.

[5]This counts the language Hindi-Urdu as separate from Khadi Boli. While, as we have mentioned, modern Khadhi Boli remains the dialect that is closest to modern Hindi, we preserve the distinction made by the website. Further, we note that Hindi-Urdu refers to Hindustani or the blend of vocabulary and other aspects from Khadi Boli with those of Farsi given the Persian influence in the Mughal rule.

Language	Folk-songs	Poetry	Folksongs tokens	Poetry tokens	Total Pieces	Total tokens
BAND 1						
Hindi-Urdu	1	54408	100	7127897	54409	7127997
Nepali	0	4753	0	692657	4753	692657
Gujarati	14	624	1795	73363	638	75158
Punjabi	754	0	69595	0	754	69595
Sindhi	0	500	0	51458	500	51458
Marathi	5	30	1412	1915	35	3327
Bangla	12	0	838	0	12	838
Avg.						1,145,861
BAND 2						
Awadhi	47	1333	4942	495137	1380	500079
Maithili	0	1552	0	218339	1552	218339
Bhojpuri	131	1275	20350	177289	1406	197639
Brajbhasha	83	1441	8883	151156	1524	160039
Magahi	340	376	37587	47167	716	84754
Avg.						232170
BAND 3						
Angika	96	6773	21419	1243727	6869	1265146
Hariyanvi	554	930	49122	183881	1484	233003
Rajasthani	67	1790	7404	180320	1857	187724
Sanskrit	2	248	184	95450	250	95634
Garwali	128	449	33380	59288	577	92668
Chattisgarhi	92	378	33504	49722	470	83226
Bhil	155	0	27326	0	155	27326
Bundeli	326	0	26928	0	326	26928
Korku	177	0	15509	0	177	15509
Nimaadi	157	0	14056	0	157	14056
Baiga	35	0	13848	0	35	13848
Malwi	129	0	9626	0	129	9626
Bajjika	0	71	0	7414	71	7414
Pali	0	27	0	5859	27	5859
Khadi Boli	42	0	4507	0	42	4507
Kumaoni	9	0	1028	0	9	1028
Bhadavari	8	0	990	0	8	990
Himachali	3	0	466	0	3	466
Kannauji	6	0	327	0	6	327
Avg.						109,751

Table 3.1: Showing crawled corpus counts for all collected languages.

```
{
  "title": "बरखा गीत / अमरनाथ मेहरोत्रा",
  "text": "\nई बदरा बरीस-बरीस के हमरा के भिंजओले हए। \nहम्मर घर ...",
  "lang": {
    "dev": "बज्जिका",
    "rom": "bajjika"
  }
}
```

Figure 3.2: File layout for stored pieces

Chapter 4

Probing the data

The data is cleaned at a character-level, by filtering out any (letter) characters that do not fall within a specified UTF-8 code-point range, as well as punctuation (both Roman and Devanagari script punctuation marks.) Tokenization is performed by simple white-space splitting.

4.1 Character-level probes

While the languages under consideration are written in the Devanagari script, a language may use certain extra characters to describe a sound perhaps not covered by the Devanagari script; alternately, some characters from the Devanagari script may not be used in certain languages. Further, since Devanagari is orthographically shallow - meaning, that the spelling of a word in Devanagari is closely representative of its pronunciation, inspecting character distributions of a language may give us a fairly good idea of the general usage of consonants and vowels in the language.

We inspect a table of (Devanagari) character distributions over the languages post-cleaning. Here, we talk about these characters transcribed into IPA. As expected, the commonest and most widely used consonants and vowels in the IA family form the bulk of the distributions of most languages, e.g. letters representing /t/, /ð/, /a/, /e/. We see some conspicuously low numbers for some characters in certain languages, e.g. letters representing /ʃ/, /v/, and /ɳ/, fairly common consonants in the rest of the languages, seem to be very little used (in this corpus) in Kannauji. This observation is in part supported by Dwivedi and Kar [2016], who say that the first two are not native to Kannuaji but borrowed from Hindi.

We also see spikes in more endemic consonants as expected, for example, the letter representing /ɭ/ - which is not used in Hindi and most of the languages under question - only shows considerable usage in Marathi and Nimaadi. Finally, the “avagraha” symbol “s”, used in Sanskrit to denote the deletion of the inherent vowel of the previous consonant, has only been inherited into the scripts of certain languages like Nepali and Magahi; in Hindi, it is sometimes used to denote the elongation of the previous vowel, especially in lyrical texts.

We calculate symmetric KL-divergence metrics for these character distributions, smoothing zero-figures if any. See Figure 4.1. Interestingly, Sanskrit and Pali (both dead languages) show the most difference from the other languages.

We notice that the eastern cluster of languages, from Brajbhasha to Angika, show high similarity to each other as well as the north-western cluster i.e. Sindhi to Bhili. Kannauji shows a rather sharp shift in similarity with the eastern languages as compared to the others.

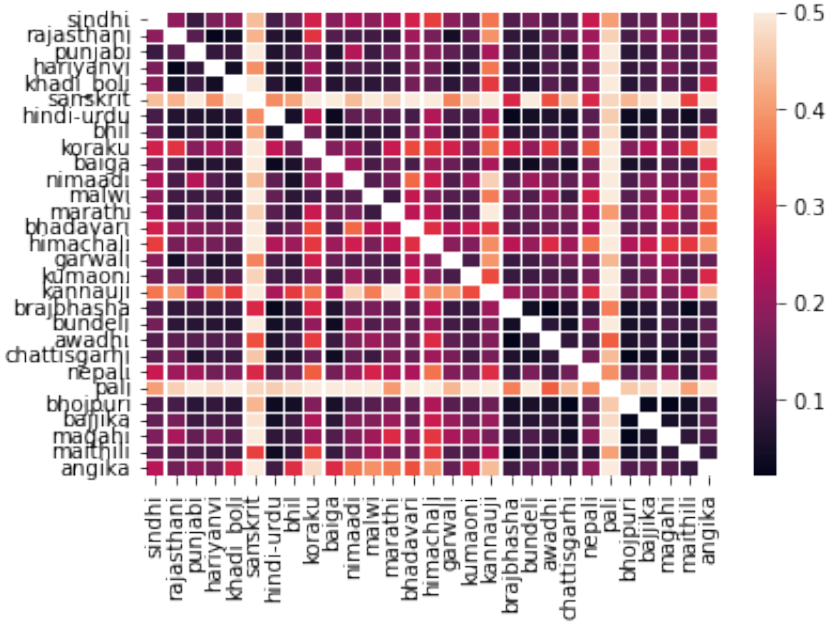

Figure 4.1: Character-level symmetric KL-Divergence for all languages

4.2 Lexical Similarity

4.2.1 Filtering

We wish to investigate the exact-match lexical overlap between each pair of languages as an initial indication of similarity. Note that since we comparing fully inflected word forms, we may miss several nearly perfect matches that differ in their inflection or due to spelling variations. We account for this in later experiments.

Given the range of dataset sizes as seen Table 3.1, we need to account for data scarcity as well as noise in larger datasets. That is, we do not wish to count a word that is regularly used in language X but appears only as a one-off occurrence in language Y as evidence of lexical overlap between X and Y. One solution is to this problem, of course, is to smoothly handle word frequencies in the two corpora by looking at measures such as cross-entropy or KL-divergence over word distributions instead of exact word counts. However, such a measure would disguise the relatively simple and interpretable quantity we are seeking to measure: that is, how many lexical items do two languages exactly share? Instead, we choose to discard all words that occur below a certain frequency in the language datasets. This threshold is naturally different depending on the size of the dataset; in specific, we discard all words in language L that occur with a frequency less than:

$$T(L) = log_{100}(N_L) - 1 \tag{4.1}$$

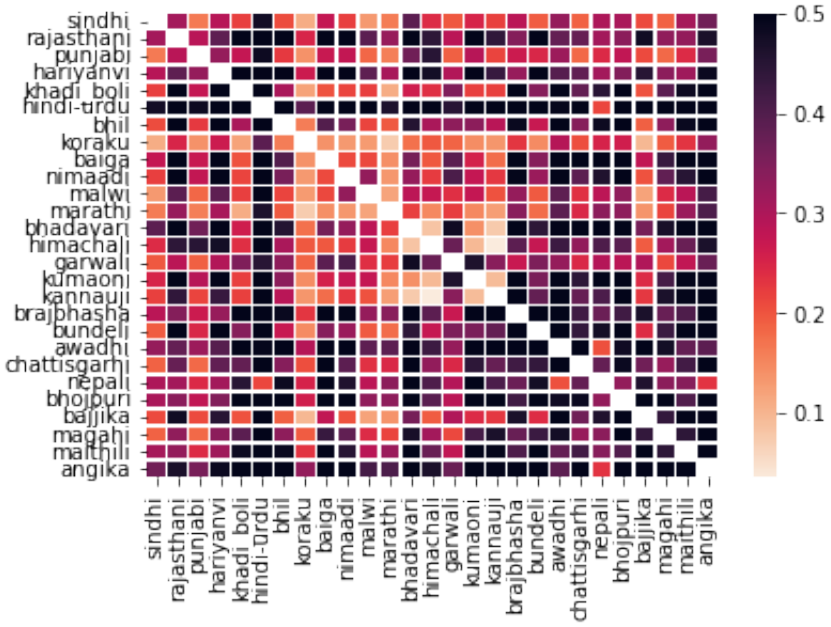

Figure 4.2: Pairwise lexical overlap for all languages

N_L is the total number of tokens in the dataset of language L. This expression roughly models the growth of the preferred threshold with the growth of the dataset size. The exponent 100 and the constant -1 were chosen to best fit the data; specifically, we wished not delete any words from languages with less than a thousand tokens.

4.2.2 Measures

Suppose L_1 and L_2 are the filtered lexicons of two languages, we want to find :

$$O_{ij} = \frac{|L_i \cap L_j|}{norm_{12}} \tag{4.2}$$

For the normalization factor $norm_{12}$, we tried both of the following:

1. $min(|L_1|, |L_2|)$: This is the natural choice; the maximum possible similarity in this case for two datasets of any sizes is 1. However it does have the following minor issue: since this denominator throws the size of the larger dataset out of consideration, it ignores the fact that a language A for which we have more data has better coverage over its own vocabulary and therefore a better chance of showing lexical overlap with a much smaller language B as compared to another language C for which we have more data than B but much less than A.

2. $|L_1| + |L_2|$: We also tried this normalization factor to hold both dataset sizes accountable, as it were. This would cap the possible lexical similarity between the two languages at $\frac{min(|L_1|,|L_2|)}{|L_1|+|L_2|}$; this naturally has the opposite issue as above; it treats languages with larger datasets unfairly.

We use the first normalization method as listed, since the benefit and easier interpretability of a $[0, 1]$ metric perhaps outweighs the discussed issue. We exclude certain languages from these experiments; namely: Sanskrit and Pali (since

they are both dead languages), and Gujarati and Bangla (since they are primarily written in different scripts). Although we leave Marathi in the mix, its results are not particularly representative - although Marathi is a Band 1 high resource language in general with a high influence on nearby languages and dialects, it has very few pieces on this website; further, many of them are from non-contemporary authors. These results, therefore, should not be taken seriously for Marathi.

4.2.3 Results: Pairwise lexical similarity

See the pairwise results for all languages in Figure 4.2. The clearest dark patch is the bottom right square, i.e. the Purvanchal and eastern languages from Kannuaji to Angika. This is expected and confirms that the corpus is representative of the close linguistic, cultural, and geographical ties between these languages.

We also see that Hindi-Urdu has high lexical similarities with almost every language. This could be the result of a few of factors: *(i)* actual linguistic closeness due to the widespread use of Hindi-Urdu, which absorbs dialectical usage as well as forming a large source for borrowings into different dialects *(ii)* the large dataset size for Hindi-Urdu, possibly including noise even after filtering *(iii)* Inclusion of non-Hindi pieces clumped into Hindi-Urdu, either because the poet wrote in several languages, or since Hindi-Urdu forms the "default" label for many works in related IA languages. The last of these is unlikely to be a large contributing factor given that the website is curated with the express purpose of documenting the literary and linguistic diversity in this continuum.

We also notice that some languages have consistently low lexical similarities with others, especially those with little data, such as Malwi. This is probably because the collected dataset is too small to be representative of the vocabulary of these languages. Finally, we observe that Korku shows very low similarity numbers with the other languages; as we have mentioned, it is not an IA language and therefore lacks the genealogical similarities of the others.

We also construct a dendrogram based on this lexical similarity measure; see Figure 4.3. We include some languages such as Gujarati and Bengali (excluded for cognate induction due to reasons mentioned) as a sanity check: we would like to see whether the measure accurately represents the fact that these languages are distant relatives of the Hindi-related languages and dialects under focus.

We see that some languages expected to be similar are grouped in the same subtrees e.g. Haryanvi and Rajasthani, {Awadhi, Angika, Bhojpuri}, as well as {Nimaadi, Malwi, Bhili, and Baiga}. More distantly related languages like Gujarati, Pali, Bangla and Sanskrit are placed on the outer parts of the tree as expected. However, we would have also expected to see Khadi Boli closer to Haryanvi, and Bajjika closer to Angika and Bhojpuri.

4.2.4 Language clusters

We also take a "close-up" look at sections of the pairwise results for language clusters that we expect to have closer relationships within the cluster. See Figures 4.5,4.4,4.6. There are 3 such geographically motivated bands that we are interested in.

Firstly, we observe the "north" band, including Sindhi, Haryanvi, Punjabi,

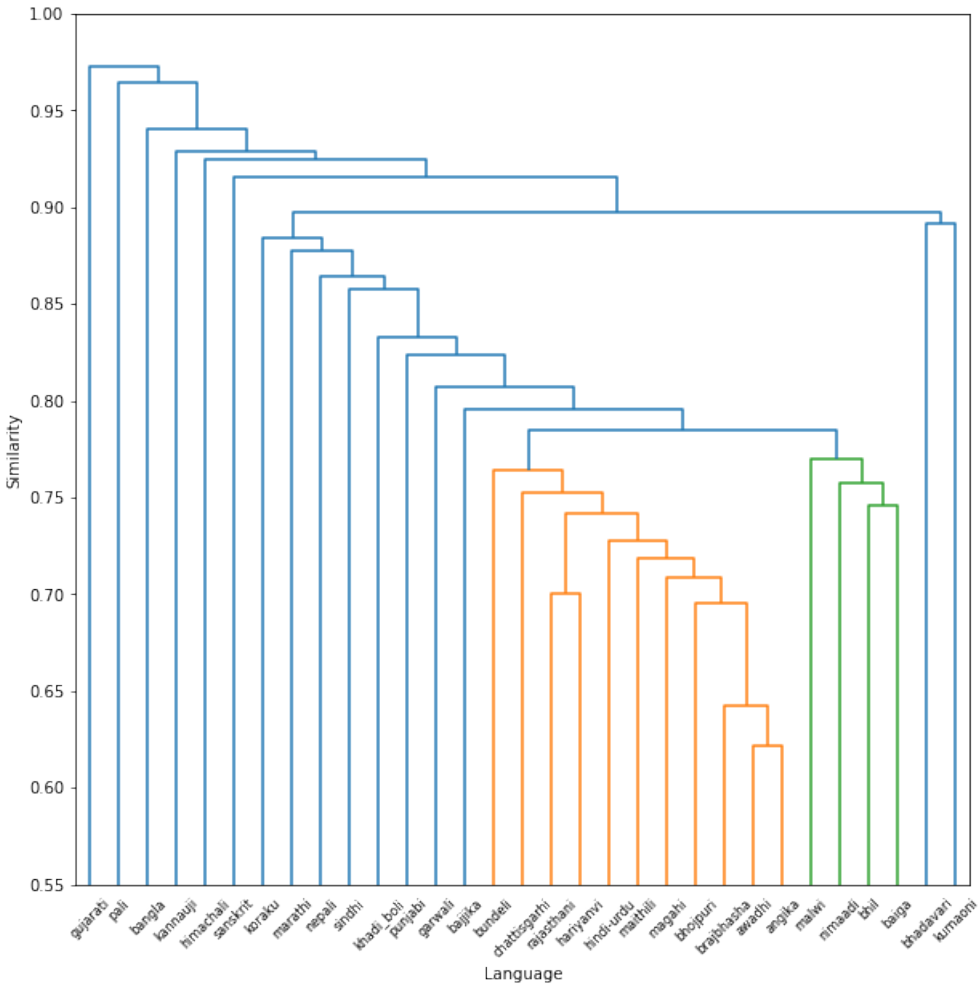

Figure 4.3: Dendrogram based on lexical similarity.

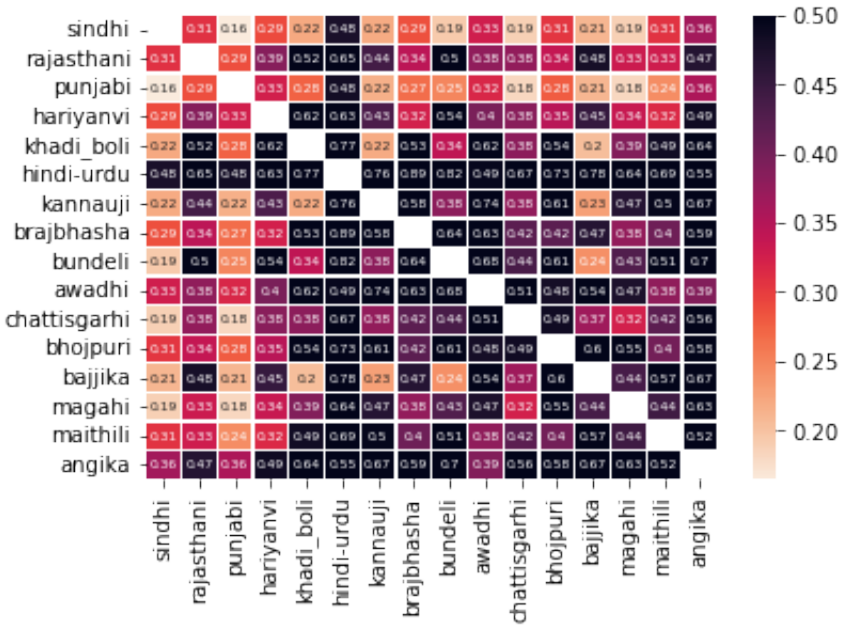

Figure 4.4: "North central" cluster of languages. Numbers in cells of this figure as well as following figures represent values of the metric being scored - in this case, the lexical overlap metric.

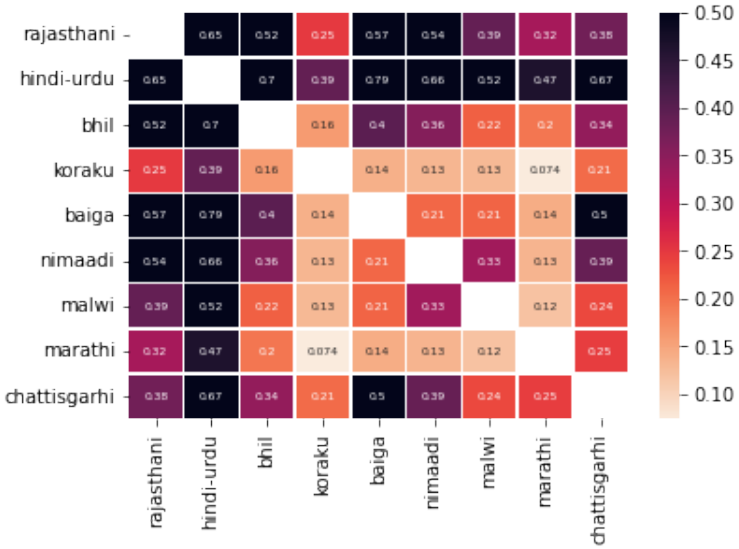

Figure 4.5: "Central" cluster of languages

Figure 4.6: "Northern" cluster of languages

and the Pahari languages. Then we have the "north-central" band, which follows the heartland of the Gangetic plains, from Rajasthan (Rajasthani) across Delhi (Khadi Boli), Uttar Pradesh (Awadhi, Kannauji), Chattisgarh (Chattisgarhi), and Bihar (Bhojpuri, Magahi, Angika). Finally, we have the "central" band across southern Rajasthan (Bhili), Madhya Pradesh (Nimaadi, Malwi) and Maharashtra (Marathi).

We see that the "north-central" band indeed has the higher inter-similarities with some pairs (even excluding Hindi-Urdu) showing similarities at around 70% (Bundeli-Angika, Kannauji-Awadhi). The "north" band follows; we see that Haryanvi and Nepali generally have high overlap with surrounding languages. Finally, the "central" band shows Rajasthani as having high lexical similarity with languages spoken in nearby regions e.g. Bhili and Nimaadi. Baiga shows generally low similarities except with Chattisgarhi, of which it is supposed to be a variant.[1]

4.3 Subword-level Probes

Since IA languages are in general morphologically rich, we also calculate some pairwise subword-level overlap measures, captured by character grams of length 2, 3, and 4. There are two main angles to these experiments; we describe each below.

4.3.1 Subword-level Overlap

Firstly, we want to capture overlap in the same way as we did for lexical similarity, thinking of subwords as approximating morphemic units of the language. Therefore, suppose L_{ic} is the inventory of c-length char-grams for language i, then we calculate c-char-gram overlap for languages i and j as:

[1] `https://glottolog.org/resource/languoid/id/baig1238`

$$O_{ijc} = \frac{|L_{ic} \cap L_{ic}|}{min(|L_{ic}|, |L_{ic}|)} \tag{4.3}$$

While calculating subword similarity over different char-grams lengths, we would like to weight O_{ijc} according to c, capturing the idea that is more of a similarity signal for two languages to share c-char-grams with greater c. For this purpose, we calculate the "universe of possibilities" for each c; i.e. the total number U_c of unique c-char-grams that occur in the entire corpus. Since we would like to have normalizing weights that are inversely related to the probability of an accidentally shared c-char-gram, we use the following expression for the final subword similarity:

$$O_{ij} = \sum_c (O_{ijc} \cdot \frac{U_c}{\sum_c U_c}) \tag{4.4}$$

4.3.2 Distributions over subwords

We also want to capture and compare languages' usage of character sequences; that is, not just the occurrence of a common particular sequence in two languages but also how frequently it is used in either.[2] This is essentially the same idea as in Section 4.1, but extended to character grams rather than single characters; accordingly we use the symmetric KL-Divergence once more to calculate this. The final metric is simply the average of the individual figures for divergence for each char-gram length.

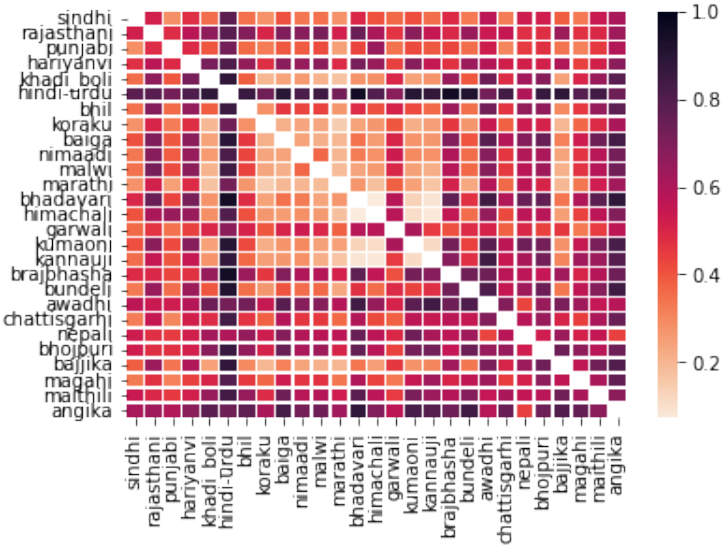

Figure 4.7: Overlap-based similarity over i-chargrams

We also tried different ranges for both of the above types of measures; in general, they follow the same trend. We only present representative heatmaps here; see Figures 4.7 and Figure 4.8 for the overlap-based and KL-Divergence-based measures respectively. As we saw in previous experiments, the eastern

[2]Remember that a sequence of characters in Devanagari is likely to correspond very closely with the way it is spoken.

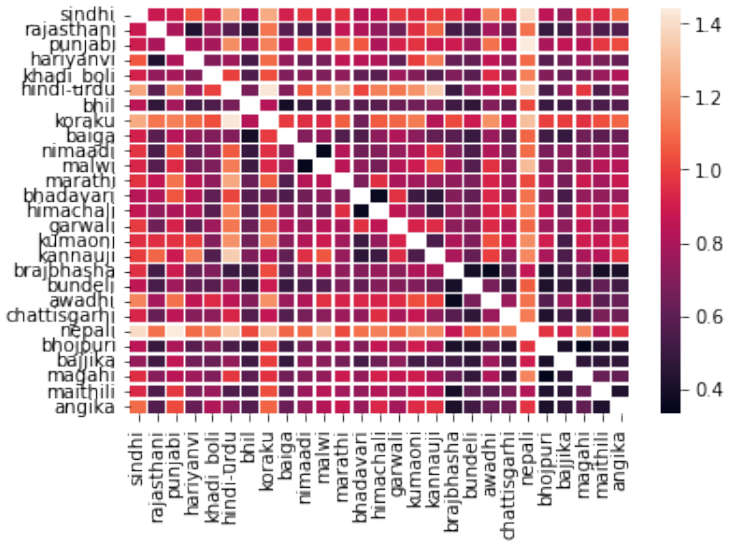

Figure 4.8: Pairwise KL-Divergence over distributions of i-char-grams. Lower is better.

cluster as well as the western cluster of languages show close relationships with each other; we have some positive outliers such as Hindi-Urdu, and some negative ones such as Korku and Himachali.

Chapter 5

Cognate Induction: Backgroud, Potential Approaches, and Baselines

5.1 Introduction: Bilingual Lexicons and Cognate Induction

A bilingual lexicon is one of the most basic bilingual resources one can have for a given language (with some anchor language). Bilingual lexicons are especially useful in low-resource scenarios to enable cross-lingual transfer of tools or resources; for example, they can be used as reference for word to word translation, or as seeds for building multilingual embeddings. This is true especially if the anchor is a high-resource language, with tools that can potentially benefit the low-resource target. In our situation, a natural choice is to choose Hindi for the anchor. Bilingual lexicons also have applications in historical linguistics, in helping linguistics to understand the evolution of a linguistic system. Finally, in the case of severely under-supported languages such as the languages under consideration in this work, they can be used for building dictionaries and aids for speakers and language learners.

A bilingual lexicon may take many forms. It may vary in its type of linguistic unit (word-form, lemma), format type (unit-unit, unit-description, and other), type of equivalence (one-to-one, one-to-many), domain, length, and so on. For traditional dictionaries used as language aids, we may have canonical word forms, or citation forms, and part-of-speech tags on the source side, and more than one possible translations including multi-word phrases and explanations, as well as examples of usage, on the target side. For computational usage in NLP or linguistics research, however, we would perhaps prefer a much simpler mapping-type structure: lemmas on both source and target side, without complex explanations or examples.

In the following approaches, we build a collection of bilingual lexicons with the following properties for every (non Hindi) target language *target*: it is in the direction **Hindi - *target***, with a **one-to-many** setup, dealing in **word forms** on both sides, with maximum length **5000**. Source words from Hindi are selected according to frequency; target side frequency constraints may or may

not be applied. The reason we work with word forms is because we do not have lemmatizers for our target languages. We use a one-to-many setup simply to increase the chances of an accurate prediction for a given source word. Finally, we use the same maximum lexicon length, implying an identical set of Hindi source words, for all approaches and language pairs to maintain comparability across approaches and languages.

Note that so far we have talked about bilingual lexicon induction, rather than cognate induction. When talking about bilingual lexicons for closely related low-resource dialects, our first target is cognates and borrowings across the languages; these are easy targets in languages with shared scripts since they tend to share a considerable part of their orthography. In this work, we do not attempt to discover lexical equivalents that are neither cognates nor borrowings, although this is an active problem in recent literature (as we mention in Chapter 2). Instead, we focus on using the relationships between these languages to build basic cognate/borrowing lexicons. We also do not distinguish between cognates and borrowings; for our purposes, both are instances of related vocabulary that come from a common origin and share some semantic and phonological characteristics.[1] Henceforth, we use the term "cognate" to include borrowings; further, the term "lexicon" or "bilingual lexicon" refers to cognate lexicons. Finally, we are also not concerned with the question of the direction of borrowing between the two languages; the origin of the borrowed lexical item may also be a third language altogether.

5.2 Background work

We give a general overview of the the literature for bilingual lexicon induction and cognate detection/induction in Chapter 2. In this section, we describe representative works in a little more detail to understand the broad kinds of approaches to the problem of cognate induction.

The field of historical linguistics has long been concerned with the question of cognate identification from semantically-aligned word lists. This is traditionally performed by trained experts, by exploiting patterns of regular sound change. Regular sound change is phenomenon by which a certain sound in one language might correspond to another sound in cognates contained by a sister language, across the map and given the correct phonological conditions. Today, automatic methods of cognate identification are being used to aid linguists in this task, for example, with the software LingPy [List, 2014].

Working explicitly with phonology/orthography: Works that deal explicitly with orthography (often transcribed into the IPA alphabet) usually deal with modelling regular sound change, similar to what linguists would do. While all the following approaches must in some way deal with orthography, these approaches work by explicitly mimicking what a linguist might do, for example, phonetic alignment (aligning what phones in two words may "correspond" to each other in case of cognacy) [Kondrak, 2000], followed by identifying cognates.

[1] In related languages with the same or similar script, shared phonologies of cognates often result in similar orthographic characteristics, as in our case. When the scripts used are different, transliteration may be used to discover correspondences, or a third script such as the International Phonetic Alphabet (IPA).

The first step can be performed with something as simple as normalized edit distance (henceforth NED), as well as more complex algorithms, whereas the second step can be (for example) performed by clustering methods that partition a set of semantically equivalent words into (cognate) clusters based on previously computed distance measures; see List [2014] for an illustration and explanation of both these steps. List [2014] also present LingPy, a library for historical linguistics that can run different algorithms within this paradigm and for related historical linguistics tasks such as constructing phylogenies. See List [2012] for the LexStat algorithm implemented in this library for cognate identification.

An evolutionary approach: Hall and Klein [2010] present an approach for identifying cognates from an *unaligned* collection of word lists of related languages, given the language family tree. This is the closest setup to ours in that we do not have semantically aligned word sets. This work uses a generative model to model three subprocesses: i.e. survival (whether the daughter language has a cognate for a given cognate set), evolution (sound changes), and alignment (which words in the observed languages are cognates with each other).

The basic idea of the algorithm is to model the evolution of words through the language family tree (note that we only have observed word lists for the leaf languages), with transducers along the tree edges. These transducers are simultaneously learnt with leaf-level alignment of cognates in an iterative manner; see Figure 5.1. The final goal is to learn the alignment parameters (i.e. which words in each language belong to which cognate set) - all other parameters are marginalized, such as reconstructed words in intermediate nodes.

MT-like approaches: In recent times - i.e. the last decade or so, different studies have applied feature learning approaches to cognate detection [Beinborn et al., 2013, Hauer and Kondrak, 2011, Fourrier et al., 2021]. These works (and many others) model cognate prediction as a low-resource machine translation task, using statistical as well as neural methods for the task of predicting likely cognates in a bilingual setup. To take one of these, Fourrier et al. [2021] compare SMT (statistical machine translation) and NMT (neural machine translation) approaches to cognate prediction, and also explore the effects of pretraining a language model on monolingual data. For SMT, they use Moses [Koehn et al., 2007], aligning the bilingual data with Giza++ [Och and Ney, 2003]. For the NMT approaches, they use two strategies: the RNN (bi-GRU) with attention [Bahdanau et al., 2014] and a Transformer model [Vaswani et al., 2017]. All approaches work at a "character" level - in specific, the words are transcribed into IPA, and segmented into phones; therefore, the unit of MT is the phone. They find that SMT still performs the best for smaller datasets, in line with the findings of previous works. They also find that multilinguality with the NMT setup, with the encoder seeing many languages, and a different decoder per language, helps to boost results; however, "pretraining" on monolingual lexicons makes no difference.

While the approaches adopted in these works hold great potential, they concern the supervised task of cognate prediction, with the smallest cognate dataset containing 1804 cognate pairs; these methods are therefore unavailable in our situation. However, it is interesting to note that machine translation is a viable perspective through which we can look at cognate lexicon induction given the very rich literature in machine translation, including unsupervised machine

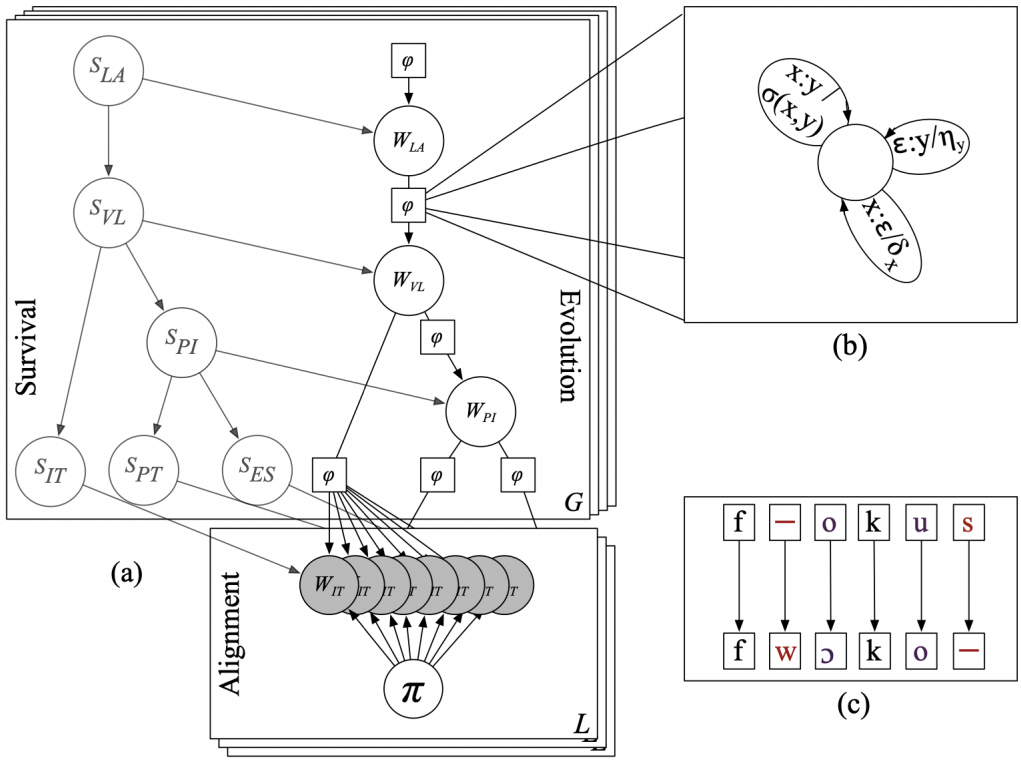

Figure 1: (a) The process by which cognate words are generated. Here, we show the derivation of Romance language words W_ℓ from their respective Latin ancestor, parameterized by transformations φ_ℓ and survival variables S_ℓ. Languages shown are Latin (LA), Vulgar Latin (VL), Proto-Iberian (PI), Italian (IT), Portuguese (PT), and Spanish (ES). Note that only modern language words are observed (shaded). (b) The class of parameterized edit distances used in this paper. Each pair of phonemes has a weight σ for deletion, and each phoneme has weights η and δ for insertion and deletion respectively. (c) A possible alignment produced by an edit distance between the Latin word *focus* ("hearth") and the Italian word *fuoco* ("fire").

Figure 5.1: Taken from Hall and Klein [2010]

translation [Lample et al., 2017].

5.3 What kind of multilingual lexicon do we want?

This section discusses our design decisions for cognate lexicons that we wish to induce, both with repsect to induction as well as with the collection of evaluation data in mind. We use the term "equivalent set" to mean a set of words, each from a different language, that are "equivalent" to each other or correspond to each other, in the sense of being cognates with a shared semantic "concept" or meaning. We use the term "lexical relationship" to mean the correspondence between equivalent words.

5.3.1 Bi- vs. multi-lingual

The phrase "multilingual lexicon" in its robust form implies that, for N languages, we should have $\binom{N}{2}$ lexical relationships for a single equivalent set. Any evaluation data we use must verify each of these lexical relationships. Clearly, this is very resource intensive, and infeasible in our current situation. In this work, we allow a "transitivity" assumption i.e. if we have (x, y) in some bilingual lexicon for languages (L_1, L_2), and (y, z) for (L_2, L_3), then we can assume (x, z) for (L_1, L_3) and list (x, y, z) as an equivalent set. We therefore use the term "multilingual lexicon" for two slightly different underlying structures, i.e.

- A collection of bilingual lexicons for N languages, for some subset of the $\binom{N}{2}$ possible language pairs. Preferably, we would have a single "anchor" language against which we have bilingual lexicons for all other languages.

- A list of N-aligned equivalent sets: for a single "concept", we have attested cognates from all N languages. $\binom{N}{2}$ lexical relationships are attested and can be inferred from each equivalent set. For example, we have a potential equivalent set p with English-French-Italian: attack-attaque-attacco, with $\binom{3}{2} = 3$ lexical relationships: attack-attaque, attack-attacco, and attaque-attacco. Each of these should be verified for p to be an attested equivalent set.

Note that the latter requires at least the former, as well as *common* words across all bilingual lexicons in order to induce equivalent sets across the languages. In our search to create a multilingual lexicon, therefore, we have greater chances of success with bilingual lexicons from one source with parallelization into N languages, as compared to isolated bilingual lexicons from different sources and probably different domains and formats; in addition, we also avoid the problem of harmonizing possibly conflicting relationships extracted from multiple sources. Using a single source with $1 : N$ format for lexical equivalents, with the transitivity assumption, is thus much more likely to facilitate the creation of a multilingual lexicon over the given bilingual lexicons, due to shared vocabulary and probably shared linguistic decisions for the presentation of word forms across all languages. The price that we expect to pay in using this format is the introduction of some bias due to the choice of the single source language, as well as noise introduced by

possible failures of the transitivity assumption we described above, when looking to infer lexical equivalents between two non-source languages.

We may go back and forth between the two above implicit structures of a "multilingual lexicon" in usage as well as creation, depending on the data available or the evaluation task at hand. However, for the most part in this work, we use the term "multilingual lexicon" to mean **a collection of bilingual lexicons, anchored against the same set of source words in a common anchor language**.

5.3.2 Format decisions

We face certain problems of principle in setting out to assemble a multilingual lexicon, or in fact, any kind of lexicon. Here are the most relevant ones for this work; we discuss each of these problems in detail below with respect to our collected lexicons.

1. What is the linguistic unit that constitutes the entries in the lexicon (on the source side)? This is directly concerned with question of dealing with the morphological richness of these languages and data scarcity both in corpora and evaluation data.

2. In the case of bilingual dictionaries, what is the format of the target side entries? That is, are they direct equivalents, or explanations including phrases and/or sentences?

3. How do we deal with synonymy and polysemy/homonymy?

Source-side linguistic unit

Lexicons may have different entry types: e.g. fully-inflected lexemes/word forms, lemmas, morphemes, morphs. In standard dictionaries for a general public, we usually see lexemes in a citation form, such as the infinitive form for verbs, and the singular nominative form for nouns. Of course, conventions regarding these defaults may differ across languages as well as dictionaries for the same language. This format is naturally intended for the ease of comprehension of lay language learners.

For lexicons collected for a computational purpose, our design decisions are mainly targeted towards dealing with morphological explosion, ease of programmatic processing, as well as ensuring a balance between complexity (for a good evaluation signal) and simplicity (to enable wider and more general usability) - constrained, of course, by the available human-labour and raw resources.

Indic languages are generally morphologically rich, to varying degrees. Further, given the nature of a dialect continuum, we expect that we will observe many shared roots overlaid with language-specific inflection paradigms, some of which can be rather extensive. This setup points to an ideal multilingual lexicon design with (linguistically motivated) morphs/morphemes as entries, although the resulting lexicon may look unnatural and be unhelpful for a language learner; not only would such a design facilitate identification of cognates but it would also allow the matching of inflection paradigms (i.e. finding equivalents for pluralising morphs, case markers, and so on).

This would, in fact, be the ideal design for a multilingual lexicon for a morphologically rich set of languages from a dialect continuum; however, in practice, the creation and usage of such a resource present several problems both for induction and in the search for gold evaluation data.

The most glaring of these obstacles is that we do not have morphological segmenters for these languages; we have, therefore, no way of going from an (evaluation) lexicon of fully inflected word forms (such as we are most likely to find in the linguistic wild), to a list of morphemes or morphs. This problem is itself is not insurmountable; we have, of course, methods of unsupervised morphological segmentation (such as Morfessor [Smit et al., 2014]) that seek to deal with exactly this situation. However, our problem is exacerbated by the scarcity of data - attempting to train such a morphological segmenter would require train and evaluation data of its own. Unsupervised morphological segmenters rely on word lists that contain adequate repeated occurrences of morphemes; this would require more data than we are perhaps capable of collecting [Žabokrtský et al., 2022]. Note that we cannot use our monolingual corpora for the purpose of training a segmenter which is then applied to the evaluation data; this necessarily contaminates the evaluation data.

Further, we would not be able to claim anything about the quality of the segmented evaluation data (since we have no segmented gold data to gauge the quality of segmentation; such an effort would require human annotation). Such an attempt would run the risk of damaging the "gold" signal under noise introduced by segmentation in our current situation of unattested segmentation methods.

For the above reasons, we decided to work with **fully inflected word forms** on both source and target sides..

Target-side entry type

Once again, we note that our decision with regard to what target-side entry type we use may be different from that made by standard dictionaries. While the purpose of those dictionaries is explaining a given word in a lucid manner, we are concerned with finding equivalents, regardless of their abstruseness. That is, a standard dictionary might prefer to explain a rare Maithili word in simple Hindi; however, we would prefer to have the equally rare Hindi cognate on the target side, or a single unit equivalent (in case of no cognacy). In general, we prefer to maintain symmetry, i.e. **use word forms** across the two sides so that correspondences between the languages is the clearest.

Synonymy and Polysemy

Ideally, we would like to allow multiple entries per language in an equivalent set (to allow for synonymy); as well as multiple entries with different target mappings for any source word if required (to allow for polysemy/homonymy in the source). Of course, this increases the complexity of such a lexicon; we therefore seek a balance regarding the number of possible entries that respects complexity as well as the need for simplicity. In this work, we allow a **one-to-many** structure for both induced and evaluation lexicons, capping the number of target side entries to 5 for the induced lexicons. However, we do not allow different sets of target mappings for a single source word (to record polysemy or homonymy).

5.4 Potential approaches

This section contains outlines of possible approaches inspired from the above literature, both as starting points for experiments in this work, as well as directions for future investigation. Here are some plausible directions:

- GIZA++ subword-level pairwise alignment
- NMT-like setup using a shared encoder and multiple decoders
- Evolutionary approach inspired from Hall and Klein [2010]

The first two approaches leverage shared phonology in the current data without positing or using the idea of shared genealogy or inheritance.

Since the Devanagari script is orthographically shallow,[2] as mentioned, we use orthography as a stand-in for phonology in the following approaches. An alternative would be automatic transcription into the IPA for the purpose of the following approaches. However, we only have such tools for Hindi; while these tools may work relatively well for related languages, we would like to avoid potential noise propagation in the pipeline.

5.4.1 Subword-level pairwise alignment

We treat corresponding/cognate words as parallel data, and use them to learn subword-level alignments in an iterative manner, with a similar approach as, for example, GIZA++ Och and Ney [2003]. As a result, we can find the probability that two words are "translations" i.e. cognates, using the distributions learnt for alignment.

We can also try adapting this idea to multilingual alignments, with the goal of simultaneously optimizing over all the alignment score over all language pairs to find a cognate set over all languages. However, we will have to ensure we are able to sidestep combinatorial explosion when looking for new aligned words in this step. An alternative is to find a flatter loss for multiple languages rather than summing pairwise losses.

The advantages of this approach include the fact that the end parameters will give us insight into phonological correspondences between languages in the continuum. It is also more or less controllable because we start with interpretable priors (such as known correspondences if any.) Associated problems include that it is highly dependant on a good seed. Further, there needs to be a good way to find equivalent sets above a threshold during search, since trying all sets is exponential in the number of languages: it is V^N given that the vocabulary size of a language is V, and therefore intractable.

5.4.2 NMT-like approach

As in above, the idea is to treat known lexical equivalents as parallel data; however, we now use and encoder-decoder setup to learn the translation parameters, similar to Fourrier et al. [2021]. In specific:

[2]While this is true also for Hindi-Urdu and Marathi, it is especially true for other languages which standardized spelling relatively recently (or not at all) and are written as they are spoken.

- We embed the input character by character and pool to produce a word embedding.
- The decoder takes this word embedding to produce a character by character output.
- The decoder can also perform attention over input character embeddings
- The final output sequence can be used as is (allowing the capability to produce unseen words and generalize over morphological endings), or re-embedded and fed into a softmax which chooses a word in the known vocabulary of the target language. That is, we can choose to have either a sequence-to-sequence or classification task (or a ranking task).

We can experiment with which parameters should be shared over languages in order to best benefit from high-resource pair lexicons and generalize over the continuum. We may also decide to always perform translation from Hindi (meaning that we use a single encoder) to other languages.

The advantage of this approach is that it is probably capable of learning more sophisticated alignments than a subword alignment tool. However, this comes as usual with decreased interpretability, both in terms of the fact that we cannot explicitly control the algorithm with prior knowledge as well the fact that post-learning, we cannot look into the model and interpret what it has learnt about the continuum. Finally, this approach requires much more data for supervision and to learn something useful than the above; it is probably not amenable to iterative learning in the manner mentioned above.

5.4.3 Evolutionary model-based

The approach explicitly uses the fact that these languages share common ancestry that resulted in cognates, as well borrowing between proximal languages.

The setup is inspired from Hall and Klein [2010] but adapted to a *wave model* of language continua. The main difference is that we will not consider a phylogenetic historical language tree, but rather a distribution of observed languages in space. The main consequence is that the "root" language is also a contemporary language (and therefore we have data for it), and that we will consider borrowings across sister languages in a horizontal fashion as well as a vertical fashion. Our constructed graph is certainly simplified in the sense that we consider only one "center of innovation" from which words are borrowed or "inherited"; this does not match the true complexity of a wave model system; however, it is not difficult to extend this to a graph with multiple such centers.

We arrange the languages in a graph, with edges flowing outwards from the "root" (or more accurately, the centre). We can set the number of intermediate nodes (for unobserved languages) as a hyperparameter; in essence, the structure of the graph is a critical structural prior. It can be inspired from known phylogenies, as well as geographical proximity. Also note that observed languages should always be leaves of the current tree (since we haven't added horizontal edges between sisters, our graph is still a tree.)

Each edge has an edit-distance based function that contains insert/delete probabilities from character to character. These are vertical edges; corresponding

closely with the setup in Hall and Klein [2010]. In this approach, we will probably consider Hindi to be the centre, since it has the most data and is a good candidate for an anchor language.

Forward pass: starting with a word at the root, traverse each edge by

- Sampling from a survive/die binary random variable (RV) along each edge. If the word survives (i.e. gets passed on), we move forwards to the next steps; else we end.
- Sample operations from the transformation matrix to produce a transformed string.
- With the final wordform in the leaf, look for the closest word from it by edit distance in the current vocabulary of the language.

Backward pass: Starting from a leaf node, we traverse upwards to the root, updating the model parameters that were used for its "evolution". At every node,

- construct the "reconstructed" posited word at that node based on its (k) children. Do this by finding an "edit-distance" mean of the k strings.
- once we have the hypothesis words at all nodes, update the edit distance transformation matrices for all edges based on the least-distance transformations from node to child along that edge. (i.e. we increase counts and therefore probabilities of whatever operations were used)

We also want to incorporate horizontal transfer as posited by the *wave model*. We do this by adding horizontal (directed) edges between (geographically) proximal languages. Each such edge has a binary "transfer" RV for whether a word in the first node is transferred (in its current form) to the second node. See Figure 5.2 for an example structure. In theory, we could allow horizontal transfer between any two languages, no matter where they are placed in the tree; in our example figure, we have restricted this to sister languages in the tree.

The forward pass and backward pass are then modified as follows:

Forward pass: if the transfer binary RV is activated, the whole word is transferred to the sister node as is, and continues a path downwards (including being transferred to other sisters). In this case, we may of course end up with more than one outcome at any given leaf. We evaluate and perform a backward pass for each of them.

Note that we need to remember at which nodes the transfer happened for any specific outcome; this can be made computationally easier and less noise-prone by restricting the "window length" of transfer.

Backward pass: If a transfer outcome is "successful" i.e. the outcome at the leaf finds a match in the target language, then we backtrack upwards vertically as usual until the we arrive at the transfer node (which we keep track of). We update the transfer variable by increasing transfer probability (e.g. by incrementing "yes" count and re-normalizing).

Advantages of this approach include its flexible transformation matrix and again, its initial and post-facto interpretability. However, it has several downsides. Firstly, this kind of model introduces many more parameters to be correctly estimated (an edit-distance cost function per edge, transfer RVs, survival RVs)

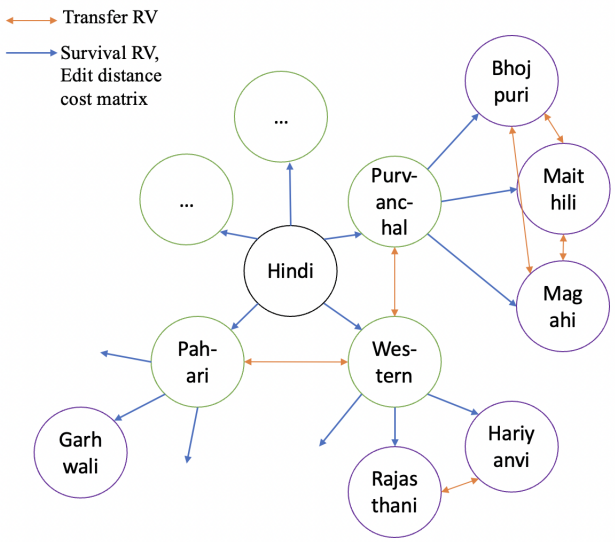

Figure 5.2: An example graph. The blue edges represent vertical i.e. tree edges, whereas the orange edges allow transfer between sister nodes. Hindi is shown as the centre here. The intermediate nodes i.e. Purvanchal, Western, etc. are genealogically motivated posited nodes that help to exploit shared relationships in certain more closely related dialects.

if we are to get accurate outputs. Once again, we require a good seed for this to work; the structure of the graph also presumably will have a high impact on the results. Finally, in an entirely unsupervised scenario such as ours, we have no way to gauge success of a hypothesised evolutionary path for a source word than crude measures such as edit distance at the leaves. This is perhaps the largest weakness of the model, making it vulnerable to noise at the crucial point of feedback.

5.4.4 Semantic spaces

Many of the align-and-cluster[3] cognate identification algorithms assume that we have semantically aligned word sets in the first place. One way to obtain such word sets is to use bi/multilingual word embeddings for this purpose. Training and evaluating these embeddings is a task in itself, since quality comes into question with low-resource settings; however, it is possible that even a noisy signal can be useful.

This idea captures an orthogonal aspect of cognacy as compared to previous approaches: shared meanings as opposed to shared phonologies with sound change. We can think of ways to use orthography-based ideas as a check to semantics-based alignments; or even semantics-based alignments as complementing evolutionary trees in order to identify matches at the leaves.

[3]Here, "align" refers to aligning phones, and "cluster" refers to clustering words into equivalent sets

5.5 Approach: Baseline

5.5.1 NED/JW

For a baseline, we use orthographic distance as a stand-in for phonological distance as mentioned before. We build a collection of bilingual lexicons for each language against an anchor - in our case, Hindi-Urdu. Source words are chosen from Hindi, and the best cognate candidate is chosen by minimizing orthographic distance. We try both normalized edit distance (NED) and Jaro-Winkler metric (JW) for the distance/similarity metric.

Implementation details: We use hyperparameters as follows:

- *Minimum Hindi-Urdu frequency*: 5. This is in place because the Hindi-Urdu corpus is indeed quite noisy as previously discussed; it is important to maintain the quality of the source words.
- *Minimum target frequency*: according to logarithmic threshold described in Section 4:
$$T(L) = log_{100}(N_L) - 1$$
- *Number of targets*: 5. This is the number of top candidates that we retrieve per source words.
- Maximum lexicon length: 5000. This means that we pick the top K (here, 5000) frequent words in the source lexicon, maintaining minimum frequency constraints. This number is not highly relevant to evaluation except for recall purposes i.e. we may cover more source words from the test lexicons if we increase this number. However, we maintain this hyperparameter through all our approaches to keep recall constant so that precision is comparable.

We maintain most of these when applicable, such as number of targets, minimum source frequency, and maximum lexicon length as constant through all our approaches to preserve comparability; in specific, the last two ensure that we have identical recall for all approaches that have the same anchor language.

Chapter 6

Approach: Expectation Maximization over orthographic score function

6.1 Introduction

A limiting theoretical deficiency in the previous approach is that it deals naively with orthographic distance, treating substitutions of any two characters equally (similarly for insertions and deletions). This is of course not what we want; we may know, for example, that a vowel-consonant replacement is much more unlikely than a vowel-vowel change. While one could attempt to solve this issue with a carefully constructed custom cost matrix for calculating edit distance,

- It is implausible to "guess" optimal scores for each operation in a roughly 50x50 matrix.
- This matrix may differ over different language pairs

The first of these issues can be mitigated by sound classes, implemented in LingPy [List, 2014], reducing the size of this matrix at the cost of detail. These sounds classes consist of equivalence classes of phones to which words are first rewritten. However, even with these, we will still have at least a 15x15 matrix. In any case, this does not address the second issue i.e. we want to apply language-pair-specific score functions.

In this approach, we try to optimize substitution probabilities iteratively while simultaneously learning cognate pairs, given two lexicons, in an expectation-maximization style algorithm. We call this approach **EMT**, EM for "Transform probabilities". A detailed explanation of the algorithm along with implementation choices is given below.

6.2 Algorithm

6.2.1 Setup

Given two word lists (that may overlap) WL_s and WL_t, we wish to learn cognates between them. Let the set of all characters of the source and target side be χ_s

and χ_t respectively. We use a scoring function $S(c_i, c_j)$, that contains a "score" for replacing any character $c_i \in \chi_s$ with $c_j \in \chi_t$; we model insertion and deletion as special cases of replacement, by introducing a null character in χ_s and χ_t as being "replaced" or "replacing" another character respectively.[1] For a given source character, S is modelled as a transformation probability distribution over χ_t. S is initialized by giving high probability (in practice, 0.5) to self-transforms and distributing the remaining probability mass equally over other characters.

The score for any pair of characters is modelled as a transition probability distribution; i.e. we must have

$$\sum_i S(c, c_i) = 1$$

We aim to learn the optimal S for explaining the cognates in the data.

Given that $C(a, b)$ is the number of times we have seen $a \to b$, and $T(a)$ is the total number of times we have seen a on the source side, our score is the frequentist probability:

$$S(c_i, c_j) = \frac{C(c_i, c_j)}{T(c_i)} \tag{6.1}$$

We also maintain a list of found cognates, so that we only update model parameters once per cognate pair. Note that a word may appear in many different cognate pairs in this setup.

6.2.2 Initialization

A good initialization is of paramount importance for such an approach. There are many possibilities for initializing C and T according to the following principles:

- The self-transform should have a high probability. That is, S should reflect that at least at initialization, it is "good" for a character to remain unchanged. This is the core assumption we make when applying orthographic distance measures to identify cognates. Of course, if a language pair overwhelmingly shows that a certain character nearly always changes to something else in the target language, this can be learnt in the course of training.
- "Similar" sounds should have a high transform probability within themselves. For example, consonants pronounced in the same place in the mouth might be expected to have a higher chance of inter-conversion.

The second of these is rather tricky to incorporate without hand-waviness. In the current run of this algorithm, we give the same initial probability to all non-self-transform operations. In specific, we initialize $T(a) = 2, \quad \forall a \in \chi_s$ (i.e. we pretend we have already seen a on the source 2 times.) If $a \in \chi_t$, we set $C(a, a) = 1$. We then distribute the remaining probability mass (i.e. 0.5) over χ_t a uniformly. We tried initializations with heavier T priors, e.g. with $T(a) = 100$, and $C(a, a) = 80$, but the algorithm has trouble learning anything new with these settings, since the prior for the self-transform is too strong.

[1] This also incidentally introduces a null-to-null operation, which does not have any meaning. This cell will be unused in the algorithm, since we only consider transformations with one non-null side.

6.2.3 Expectation step

In the expectation step, we are trying to find new cognates given current model parameters. Given a candidate source and target pair (s, t), we can find $Ops(s, t)$, which is the *minimal list* of the operations we need to perform to get from s to t; therefore, each member in Ops is of the type (c_i, c_j). In addition to "insert", "delete", and "replace" operations, we also use a "retain" operation, for characters that remain the same in source or target. This is because we also want to estimate $S(a, a) \forall a$ to get a good estimate of the distribution over target characters for a given source character.

The score for the pair (s, t) (lower being better) is computed as

$$\zeta(s, t) = - \sum_{(a,b) \in Ops} log_{10}(S(a, b)), \tag{6.2}$$

where the lower the ζ the more probable a pair is a cognate. For a given s, we can then always find the word that is the most probable cognate as $t = argmin_{t_i \neq s}(\zeta(s, t_i))$.

Note that in the training phase, we disallow $s = t$, to mitigate exploding self-transform probabilities. This artificial restraint should perhaps be softened; however, in the current run we still have this as a hard constraint.

Finally, we choose the best K of all cognate pairs i.e. those with the highest confidence, equivalent to the lowest ζ-values. These cognates are used for updating the model parameters.

6.2.4 Maximisation

In this step, we update model parameters based on newly identified cognates in the previous step. This is performed by increasing the counts of all observed edit distance operations. That is, we set:

$$C(a, b) := C(a, b) + 1 \quad \forall (a, b) \in Ops(s, t)$$

$$T(a) := T(a) + 1 \quad \forall (a, b) \in Ops(s, t)$$

Remember that a or b here may be the null character in case of an insertion or deletion.

6.2.5 Building a lexicon

We build a lexicon by selecting each source word and choosing the best target candidate by minimizing $\zeta(s, t)$ as described above. The main difference from training is that we now allow identical word matches as candidates.

6.2.6 Hyperparameters

These are the hyperparameters we use for the above algorithm:

- $iterations = 500$
- $updates = 10$: This is the number of best scoring word pairs we choose per iteration to update the model parameters.

- $batch\_size = 100$
- $init\_total = 2$: This is the total number of counts we "distribute" over transducer operations as a prior.
- $init\_self\_count = 1$: This is the count given to the self-transform; we initially have 50% weightage given to the self-transform, and the remaining probability mass is equally divided among the other operations.
- $minimum\_source\_frequency = 5$
- $minimum\_target\_frequency$: This is decided by the threshold given in Equation 4.1 we have used for other experiments.

6.3 Pitfalls of this approach

There are a few problems with the above approach:

- Ideally, it requires a good seed that can set the EM process on the right track. Note that the seed cannot be identical words - the algorithm does not learn anything from identical word matches. Unfortunately, we do not have a good seed, either in terms of a weight matrix or with pre-existing word equivalents.
- It only deals with single character substitutions. We hope that longer equivalences will be learnt over multiple iterations; however, this is certainly a problem with the current implementation.

The latter problem can be dealt with some additional work in the following manner: Suppose we want to work with bigrams in addition to unigrams. We construct a new bigram alphabet, consisting of all possible bigrams as individual characters (for example, we can map bigrams to integers). When updating our weight matrix, we can now count two consecutive edit distance operations as a bigram change, and use our bigram alphabet as a valid entry in the weight matrix. However, in this work we only experimented with a simple unigram weight matrix.

Finally, the primary flaw of such an approach when it comes to cognate induction is that it pays no heed to the meanings of words; this ignores an essential characteristic of cognates i.e. they mean similar things in the two languages. Our next approach attempts to take this into account.

Chapter 7

Combining weak semantic and phonological signals

7.1 Idea and Algorithm

We mentioned in the previous chapters that orthographic matching, even with tailored and learnt substitution matrices for a given pair of languages, may be inherently incomplete is because it ignores the semantics of candidate words. In our next idea, we use bilingual subword embeddings to address this problem; that is, we use the semantic space to narrow down possible candidates, and then use orthographic matching in order to select the top K candidates for a given source word a.

We think of this as a two-stage approach that relies on two separate metrics: firstly, the quality of semantic similarity judgments provided by a semantic embedding space, and secondly, orthographic similarity judgments provided by the distance/similarity metric we choose to use. We optimize these two stages separately; they are described below.

7.2 Training embeddings: JOINT

Providing static embeddings for these language is in itself a valuable outcome. The main obstacle to training word embeddings for these languages is the same as before, i.e. data scarcity for most Band 3 languages makes it unlikely that a trained space will be of high quality. While it seems that we can exploit the shared genealogy of these languages to train bilingual embeddings, leveraging the relatively abundant resources of Hindi-Urdu, it is still unclear as to whether bilingual transfer of embeddings can be achieved with drastic data asymmetries.

There are several possibilities, across different dimensions, for building a semantic space for our data. Here are some questions we must answer:

- Should we build work with bilingual embeddings or multilingual embeddings? On the one hand, it will be simpler to deal with evaluating bilingual spaces, and applying them to bilingual lexicon induction for each of our languages separately; on the other hand, the languages may benefit from shared knowledge across all languages, given that they are all related.

- If we are working with multilingual embeddings, should we work with subsets of related languages? We see some definite groups within our 26 languages, e.g. the Purvanchal languages (Bhojpuri, Magahi, Maithili) seem much more interconnected at a subword level than with other languages, and so on. Perhaps it may be beneficial to use these subgroups to train different multilingual spaces to reduce noise from less related languages and maximise gains from related languages.
- How can we deal with data asymmetry when working with Hindi-Urdu as the anchor in a bilingual space?
- What sort of model should we work with, i.e. static/contextual embeddings, word level/subword level embeddings, Word2vec, GloVe, or other models?

Of course, it would be difficult to explore the entire space of these possibilities. For initial experiments, we train bilingual static embeddings for each target language with Hindi-Urdu. We use fastText [Bojanowski et al., 2017] for this purpose, hoping to leverage its usage of subword level information, given that that we are dealing with data-scarce morphologically rich languages. We use 300 dimensions, although a lower number may result in better representations for lower-resource languages, and we use a minimum corpus frequency threshold of 5.

7.2.1 Improving embeddings: UPSAMPLE

One of the problems with the above JOINT setup is the large disparity in the amount of data used for each language in the joint approach. In specific, we are applying the same minimum frequency threshold (that a word must have to be embedded by the model) for both languages by mixing the data: this threshold is more suited to the high resource language. Clearly, this is unfair to the target language data.

In order to mitigate this problem, we oversample the target language data to bring it to the same order of magnitude as the Hindi data; by artificially multiplying the frequencies of target language words, we ensure that a fairer threshold is applied to them. Note that of course this is not a real solution to the data disparity; after all, the Hindi language words are seen by comparison in many more different contexts and therefore have a better chance of being modelled correctly. However, we hope to address the basic problem of frequency thresholds by this trick, and hope that once the target language words are seen, we will observe better results. In addition, we also reduce the number of dimensions of the model to 150.

7.2.2 Visualizations

We use TSNE [Gisbrecht et al., 2015] to obtain the visualizations for JOINT models of Hariyanvi, Bhojpuri, and Rajasthani, Magahi, and Korku (with Hindi-Urdu). We present the Hariyanvi visualizations in Figures 7.1; see Appendix A for visualizations of the Bhojpuri and Rajasthani spaces.

The main observations we can make for this type of model, common to all the plots, is that the low-resource target language words seem to be clustered around

each other, whereas Hindi-Urdu words and words belonging to both languages are better situated according to their semantics. For example, in the Hariyanvi-Hindi/Urdu space, we see some unrelated words very close together, such as the bottom left cluster containing words for "business", "king", "pot", "change", "no" - a seemingly themeless cluster of words.

We also see that despite using frequency thresholds for labelling the language of the words, the "both" category is noisy and inflated, containing words that may not belong to either of the languages despite appearing above the frequency threshold.

The isolation of the target language words as observed above may have a couple of reasons. It may be that the joint training we employ is simply not good at capturing cross-lingual similarities. However, it seems that shared vocabulary and subwords across the two languages should counteract this and push towards a more integrated space. Another possible diagnosis is an effect pointed out by Gong et al. [2018]; this paper shows that low-frequency words tend to cluster together regardless of their semantics. Due to our data-assymetric situation with low frequency LRL words and high frequency Hindi words, this scenario is directly applicable to our training setup.

In response, we trained the UPSAMPLE models as described above; we visualize the same words per language for the same three languages i.e. Hariyanvi, Bhojpuri, and Rajasthani. As before, see Figure 7.1 for the Hariyanvi plot, and Appendix A for the Bhojpuri and Rajasthani plots. While it is not clear from the visualization that the UPSAMPLE models are less language-wise clustered than the JOINT, the target language words seem at least much better distributed, and we see more meaningful collocations (both monolingual in the target language, and cross-lingual) that we did not see before. For example, we see:

- "pot" (Hariyanvi), "fill" (Hindi)
- "change" (Hariyanvi), "change" (Hindi)
- "imagination" (Hariyanvi), "mind" (Hindi)
- "vedas", (Hariyanvi), "words" (Hindi), "text" (both)

However, it is difficult to say from such visualizations which space is better embedded.

7.2.3 Tests and evaluation

Integration

We would like to evaluate the bilingual quality of our embedding spaces; however, we do not of course have labelled test sets usually used for this purpose, such as word similarity datasets containing annotated similarities for word pairs, or gold bilingual lexicons. We use a rough heuristic *cl_integ* as a measure of how well the two language words are collocated. To calculate this measure, we simply sample from a source language lexicon, find K nearest neighbours of each word in the sample, and calculate the macro percentage of all such neighbours that belonged to the other language (either exclusively or as a shared vocabulary item). We calculate this both ways i.e. with either language as the source. See Table 7.1 for the results for the JOINT and UPSAMPLE models.

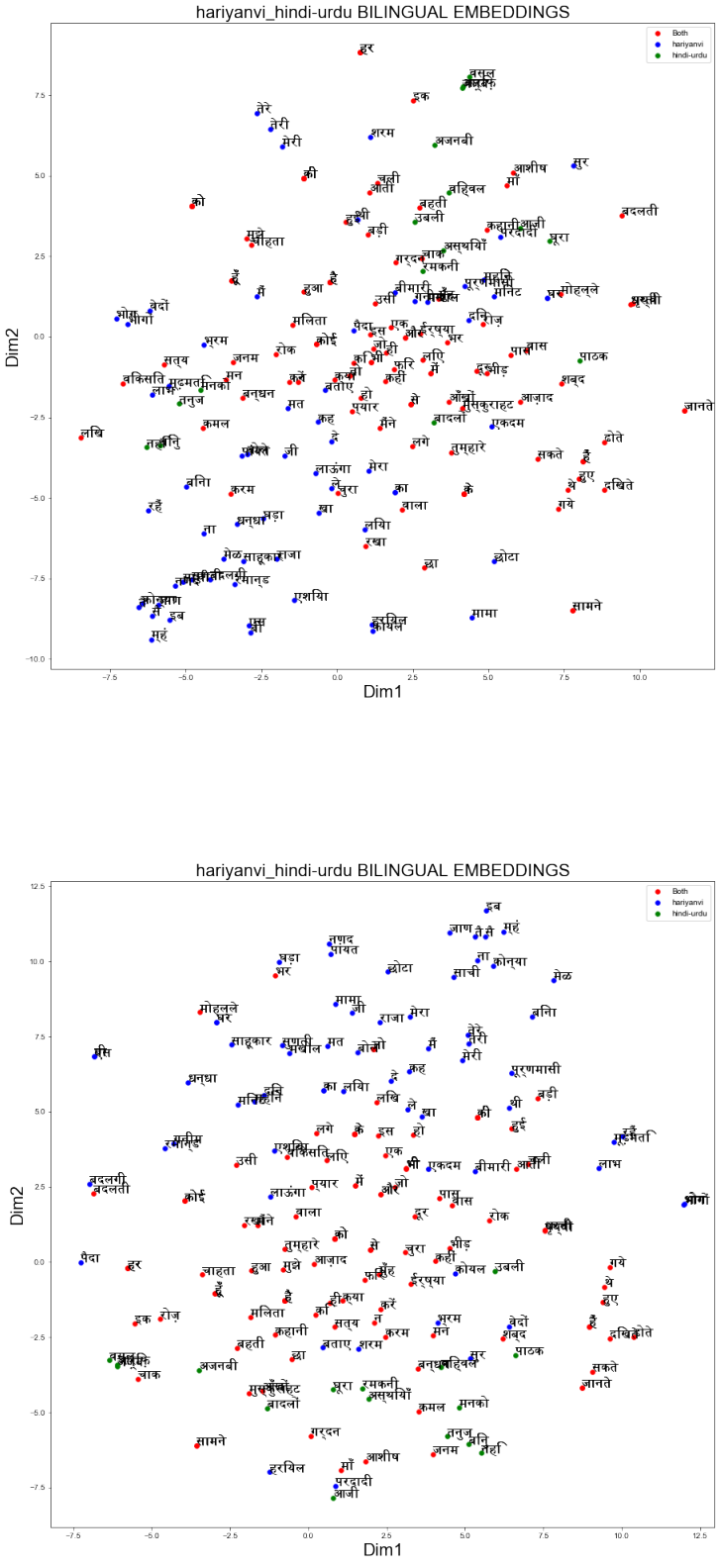

Figure 7.1: Visualization of Hariyanvi-Hindi bilingual space, JOINT (up) and UPSAMPLE (down)

	J_12	J_21	U_12	U_21
sindhi	0.53	0.23	0.31	0.33
rajasthani	0.78	0.33	0.62	0.40
punjabi	0.58	0.19	0.40	0.27
hariyanvi	0.75	0.30	0.66	0.36
khadi_boli	0.99	0.18	0.76	0.13
sanskrit	0.33	0.28	0.12	0.26
bhil	0.92	0.24	0.53	0.34
koraku	0.59	0.13	0.34	0.10
baiga	0.97	0.21	0.73	0.31
nimaadi	0.87	0.16	0.47	0.21
malwi	0.88	0.14	0.45	0.13
marathi	0.95	0.20	0.32	0.15
bhadavari	1.00	0.12	0.81	0.30
himachali	1.00	0.07	0.48	0.07
garwali	0.64	0.25	0.25	0.39
kumaoni	0.97	0.09	0.74	0.05
kannauji	1.00	0.04	0.66	0.14
brajbhasha	1.00	0.32	0.74	0.38
bundeli	0.99	0.21	0.58	0.36
awadhi	0.69	0.34	0.45	0.43
chattisgarhi	0.86	0.29	0.51	0.36
nepali	0.37	0.39	0.31	0.48
pali	0.57	0.11	0.07	0.10
bhojpuri	0.91	0.32	0.74	0.41
bajjika	1.00	0.20	0.74	0.30
magahi	0.84	0.21	0.44	0.42
maithili	0.85	0.38	0.57	0.49
angika	0.63	0.44	0.50	0.40

Table 7.1: *cl_integ* values reported as 0-1 measure for both sets of embedding spaces, in both directions. The suffix "12" indicates that we consider the non-Hindi language as source, and look for the fraction of nearby Hindi words, "21": vice versa.

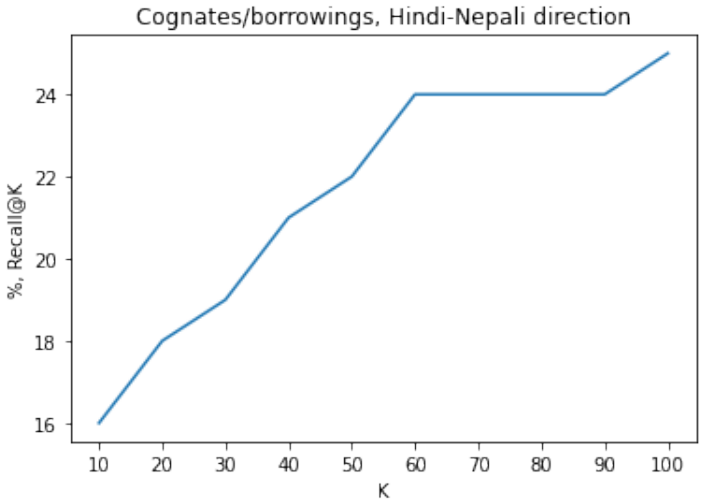

Figure 7.2: $Prec@K$ for the bilingual fastText Nepali embeddings

Nepali Bilingual Lexicon evaluation

As mentioned before, we do in fact have WordNets from the IndoWordNet project [Kakwani et al., 2020] for Nepali and Marathi, from which bilingual lexicons with Hindi can be extracted. While the Marathi dataset in our current collection is small and not very representative as previously discussed, we evaluate the Nepali-Hindi bilingual space using the Nepali WordNet. We used the WordNet to extract a Hindi-Nepali bilingual lexicon, and we calculated $Prec@K$, in the following way:

For each Hindi-Urdu word, we extract its K nearest Nepali neighbours. If any of those are the gold target, we count a full point for that word. Finally, we report the total such points as a percentage of the length of the gold bilingual lexicon.

Note that recall is 100%, since any Hindi word can be embedded with the fastText embedding space; $Prec@K$ measures what percentage of words are accurately mapped to a Nepali equivalent, when retrieving K nearest neighbours for every query.

See the results for the JOINT Nepali model in Figure 7.2, showing how precision varies with K.

Nepali is in the highest range of availability in our current dataset, so we do not expect these results to be representative for other languages with less data. We therefore also look at these results over artificially smaller cuts of the Nepali dataset. See Table 7.2. We also report these numbers for the UPSAMPLE Nepali model in the same table.

7.2.4 Discussion

There are a couple of interesting things to note about the above results. We see that *cl_integ* shows high values from the LRL to Hindi/Urdu direction, but not vice versa. Nepali happens to be an outlier in this case, which is unfortunate since it is unlikely to be representative of the other languages, and it is the only language we can evaluate with more detail.

We notice in Table 7.2 that the results for the WBST bilingual lexicon test

# tokens	cl_integ_12	cl_integ_21	bl_12	bl_21
JOINT				
5000	0.43	0.37	0.30	0.21
50000	0.33	0.38	0.29	0.21
100000	0.29	0.37	0.29	0.20
500000	0.33	0.44	0.29	0.20
UPSAMPLED				
500000	0.29	0.42	0.33	0.15

Table 7.2: Results for $K = 50$ for Nepali data splits of different sizes. 12: Nepali as source, 21: Hindi as source. *cl_integ* test checks integration of the two languages, in both directions, *bl* shows results on the bilingual lexicon test against the Nepali WordNet. We also show results for *cl_integ* and the bilingual lexicon test for the UPSAMPLE Nepali model

seem to be stable across different data splits, and if anything, decreasing with more data. This is rather suspicious; however, a possible explanation is that the positives accrue from frequent words anyway, present in all splits and possibly also present in the Hindi-Urdu data; therefore, reducing the number of Nepali tokens does not seem to affect this number. Note that this is not at all an indication that the resulting embeddings are of the same quality across different splits, simply that this metric is not able to capture possible underlying damage.

The main observations regarding the UPSAMPLE models are:

- The visualizations show the target language words to be better distributed
- The *cl_integ* values seem more uniform for both directions, and higher for the Hindi-$< target >$ direction (which is the relevant one for our approach)
- The Nepali WBST shows better recall (for all data) for the UPSAMPLE model in one direction, but worse in the other direction.

These are good indications that upsampling did indeed improve the quality of the bilingual embedding space.

7.3 SEM_JW: Semantic similarity with Jaro-Winkler

In this approach, we retrieve K nearest neighbours of each source word. These candidates are scored by an interpolation of orthographic distance and semantic similarity. In specific, we use JW for the former, and cosine similarity for the latter. We use $K = 50$; we want to have a large dragnet in order to increase the probability of "catching" the accurate map. Note that all words that are not within the K nearest neighbours are discarded from consideration no matter their orthographic similarity score. The idea is to mitigate the effect of chance orthographic similarities.

For all target candidates (i.e. the nearest neighbours), we minimize:

$$D(a, b) = 1 - scos(a, b) \cdot J(a, b), \tag{7.1}$$

where $scos(a, b)$ captures the cosine similarity between the respective vectors scaled to [0, 1], and $J(a, b)$ is the JW similarity.

7.4 SEM_EMT: Semantic similarity with EMT

Finally, we seek to combine the benefits of iteratively learning an edit-distance cost function with those of using semantic spaces. This approach is almost identical to that in Chapter 6, except for the fact that only K nearest neighbours of a source word in the semantic space are used as potential cognate candidates for that source word, both during training and inference; these candidates are then scored as usual using the learnt transform probabilities. We use the UPSAMPLE set of embeddings spaces, and $K = 50$ as before. All EMT-related hyperparameters are the same as mentioned in Chapter 6.

Chapter 8

Collecting Evaluation Data

8.1 Introduction

As mentioned before, our 16 Band 3 languages are zero-resource; this implies not only that we have no previous collection of monolingual data available for the NLP community, but also that we lack any evaluation resources for cognate induction (or any other task). We accordingly seek to collect evaluation data ourselves in the form of gold parallel lexicons for as many of these languages as possible.

Like corpus collection, this is a challenging task due to the paucity of online material and previous research in Band 3 languages. That is to say, such a "gold" lexicon must be created from human-annotated material; however, there are no pre-existing curation of lexical equivalents across our 25 languages. In this work, we hope to adapt information found on the web to our purposes, preserving its "gold" quality as best as possible. The data thus collected will be used solely for evaluation.

8.2 Existing resources

8.2.1 Looking in the wild

For some Band 1 languages (specifically, Hindi, Nepali, and Marathi), we have WordNets from the IndoWordNet project [Sinha et al., 2006, Debasri et al., 2002], from which we can extract equivalents across languages. We are not concerned, therefore, with searching for lexical resources for Band 1 languages. For some Band 2 languages (Bhojpuri, Magahi, and Maithili), WordNets are under way [Mundotiya et al., 2021] but as yet unavailable.

For Band 3, as discussed, we do not have any pre-existing bilingual or multilingual lexical resources in a convenient format. We therefore look for bilingual lexicons in the "wild"; that is, blogs, websites, scanned dictionaries, etc. We list all such raw material that we found that could be potentially useful for this purpose in Table 8.1.

The names of these resources are listed separately in Table 8.2.

We exclude a few other resources we found due to too small a length (< 30), or too unstructured a format, as unlikely to be of much help to computational linguists.

Page	Languages	Anchor language	Content notes	Format	Approx. length
1	Rajasthani^{r}	Eng.^{r}	Explanations in English	Simple list	>500
2	Rajasthani^{d}	Hin^{d}, Eng^{r}	Hin equivalents, Eng explanation	Webpages by initial letter	> 500
3	Angika^{d}	Hin^{d}, Eng^{r}	Explanations	Each word on diff. page, disabled copying	102
4	Bundeli^{d}	Hin^{d}	Equivalents	Simple listing, disabled copying	Few 100s
5	Haryanvi^{d}	Hin^{d}	Equivalents	Simple list	< 100
6	Chattisgarhi^{d}	Hin^{d}	Explanations	Webpage per word, disabled copying	< 100
7	Chattisgarhi^{d}	Hin^{d}	Equivalents	List, disabled copying	Few 100s
8	Kumaoni$^{d\ r}$	Hin^{d}, Eng^{r}	Equivalents, categorized by themes	Simple list	< 100
9	Brajbhasha^{d}	Hin^{d}	Equivalents/ explanations	Mixture of paragraphs and lists, rather disorganized	Few 100s
10	Bhojpuri^{d}	Hin^{d}	Mostly equivalents, also Hindi synonyms	Simple list	400
11	Hin.^{r}, Mar.^{i}, Nep.^{i}, "Bihari"^{i}, Mag.^{d,i}, Marwari^{i}	-	Cognates	Swadesh list	207
12	{Bhoj., Gar., Hin., Mar., Nep., Mag., Mai., Sin.}^{d,i}	Eng^{r}	Short phrase translations	Simple list	45 phrases (on avg.)

Table 8.1: Raw resources found for different languages. The superscripts ^{d}, ^{r} and ^{i} indicate that the script used for the language is Devanagari, Roman or IPA respectively. The length given is an approximation because some of these formats make it difficult to get the exact number of entries.

Page	Name
1	Rajasthani Language Dictionary — Rangrasiya
2	Glossary of Rajasthani Language - Jatland Wiki
3	Angika Shabdkosh
4	Bundeli Shabdkosh
5	(Blog post) Learn Harayanvi Language Through Hindi Language
6	Chattisgarhi-Hindi online dictionary
7	(Post) HS MiXX Entertainment
8	Kumaoni Boli
9	(Blog post) Learn Brajbhasha Vocabulary
10	(Blog post) Bhojpuri dictionary
11	(Blog post) Swadesh Word List of Indo-European languages
12	Omniglot

Table 8.2: Resource websites

8.2.2 Overview of existing resources

All listed resources together cover 4 Band 2 languages and 7 Band 3 languages: this is counting "Bihari" as the same as Bhojpuri, and Rajasthani the same as Marwari. (Note that these resources may cover more languages; we have only listed the ones relevant to this project in the "Languages" column.) However, these resources have widely different domains, content types, and formats.

Four of the listed websites disable copying and sometimes webpage inspection, discouraging crawling or re-using their data, and rendering 3 Band 3 languages once more resource-less. Content-wise, we see that many resources have explanations on the target side (Hindi or English), rather than equivalents. For this project, that means that the resource is not really ready-to-use as a bilingual lexicon, but will require further work in terms of extracting equivalents from the explanations for the target side. Resource *1* for Rajasthani also requires transliteration for the source side before it is useful. Finally, we note that even the resources listed as containing equivalents in Table 8.1 usually contain a mixture of equivalents, explanations, and examples. That is, each resource would require considerable processing, possibly manual, to yield a usable set of consistently formatted bilingual lexicons.

As we discussed, for the purposes of this project, we would like to have not only bilingual lexicons per language with an anchor (preferably Hindi), but also considerable intersections between the lexicons to allow the potential of testing multilingual interactions beyond Hindi-*lang* tasks. This too, unfortunately, is likely to be a problem when gathering resources from different sources with rather small lists.

We decided not to attempt gathering lexicons from these different resources for individual languages with the intention of putting them together, due to

the above problems, including potential extensive manual efforts to the above individual resources usable, probable multilingual mismatch, and low coverage of Band 3 languages. Instead, we look for a source that is more multilingual in its scope.

11 is naturally exactly what we would have liked to find, although, again, it may require transliteration from IPA from most languages to be useful (and for Hindi, from a "casual" Roman script). The main problem, however, is that it deals with 3 Band 1 languages (for which we already have lexicons), 2 Band 2 languages, and only 1 Band 3 language, making it a low-coverage resource for our situation.

12 is another interesting multilingual resource, highly similar to the resource that we finally decided to use but with lower coverage.

Note that a few of these resources are valuable on their own, e.g. *10* for Bhojpuri is extensive, simply formatted, and relatively neat and consistent; it will not require too much manual work to convert it into a usable resource for linguists. Similarly, *1* and *2* in Rajasthani provide the raw material for good bilingual lexicons, although they will first require a good quality transliteration into Devanagari for the Rajasthani side.

8.3 "Languages Home": Website

8.3.1 Introduction

`https://www.languageshome.com` is an online language learning website, with translations of artificially simple sentences into 76 Indian languages (including Dravidian and other languages), and some foreign languages such as French and Italian. Of these, 21 languages are of our interest, including Hindi, Marathi, Sindhi and Nepali, all 5 Band 2 languages, and 12 Band 3 languages. This resource, therefore, has considerable coverage, more than what we would be able to achieve by putting together resources for individual Band 3 languages. Its structure is similar to that of *12*; however, it is better both as regards coverage of languages, as well the fact that it has almost double the number of parallel sentences per language. These are the reasons that we chose to work with this resource over any of the other listed resources in Table 8.1, individually or in combination.

8.3.2 Format and Script

The website is arranged in a straightforward manner: each language has a separate webpage, that lists sentences in the anchor language followed by a translation in the target language. Most of our languages of interest are only anchored against a set of English sentences.

The script used is entirely Roman (for our webpages of interest): the English sentences uses standard Roman spelling, and the target Indic languages are transcribed in "casual" Roman transliteration. That is, words in the target language are transliterated "by ear", without standardization across languages or even within the same language, and using no recognizable scheme (such as ITrans or WX notation). The Devanagari spelling or the pronunciation of the

word cannot be recovered from this transliteration unambiguously without first guessing the intended Indic word. For example, both the long and short vowels /iː/ and /i/ are sometimes transcribed as "i", whereas they are distinguished in Devanagari. This is also true for other vowels pairs differing in length. Similarly, the transcribed "a" might refer to the schwa (inherent to a consonantal character in Devanagari) or the vowel /a/. The transcribed "n" may refer to nasalization of a vowel or the alveolar nasal consonant /n/ or its retroflex counter /ɳ/, all of which are distinguished in Devanagari.

8.3.3 Content

The sentences and phrases are artificially short, intended to be illustrative to language learners. For example, "He ate an apple", "He will come", "He will go". There are also single word translations for pronouns and common verbs such as "go", "run", and others. However, the website is by no means exhaustive over possible verbal or other inflections.

In total, there are about 80-90 such parallel phrases per language. They are mostly identical over different languages, with some languages missing a few, or containing minor variations.

Besides script-related noise, the sentences contain lexical noise, i.e. departing from a strict translation. This may be in the form of alternate translations separated by a "/", inflectional forms explained in parentheses (such as mentioning gender inflection for verbs), or other such information. Here are two example pairs verbatim from the English-Hindi set:

```
English: Open
Hindi: Kholo/ Kholiye (respect)/ Kholna

English: Opened
Hindi: Khola (he)/ Kholee (she)/ Khole (plural)
```

"Open" is provided with three translations: the imperative "Kholo", the imperative honorific "Kholiye", and the infinitive "Kholna" (to open). In the next example, the inflectional endings of the participle are explained by the parenthetical "he" and "she" for gender, and "plural" as applied to a plural object.

Such markings are inconsistently provided over languages and within a single language. We also note that male pronouns and inflections are considerably overrepresented as compared to female ones. Finally, we see that as with any pair of parallel sentences, all equivalents to a given word in the source are (naturally) not represented in the target and vice-versa. That is, the authors have chosen a single translation, and - unmotivated by linguistic considerations - this choice may not be the cognate of the source word even if a cognate does exist in the target language. The chosen target lexical equivalent may even be code-switched English instead of a word in the Indic language, if code-switching is common in that context. For example, we see in :

```
English: Will you give me your pen?
Hindi: Kya tum mujhe apna pen doge?
```

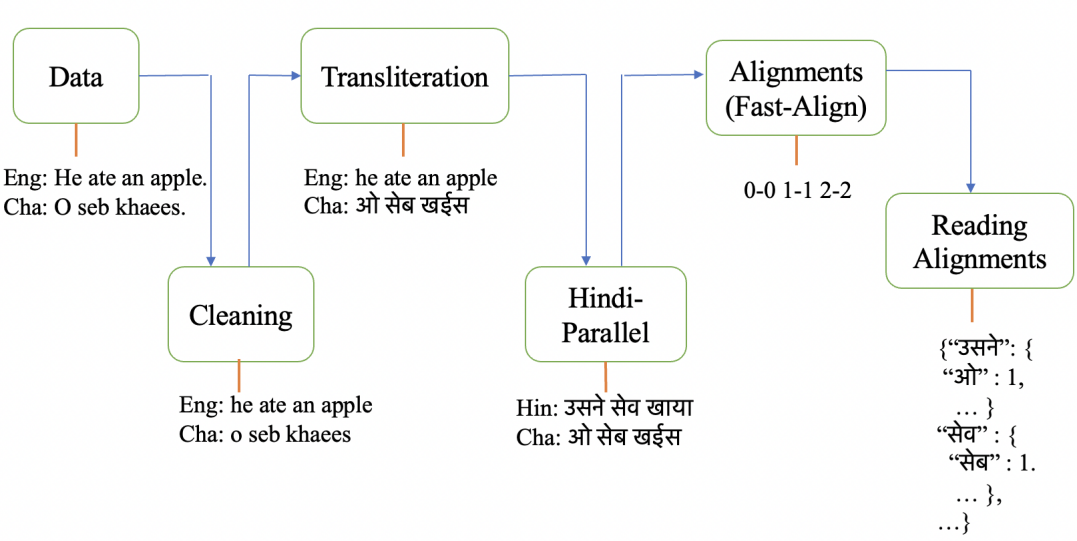

Figure 8.1: Pipeline processing raw evaluation data into one-to-many lexicons. This is a simple example with word-for-word equivalents in Hindi and Chattisgarhi.

that the word “pen” is code-switched in Hindi, rather than using the Hindi word “kalam”. However, in other languages such as Bagheli, we see the word “kalam” used instead.[1] Therefore, although the word “kalam” exists in both languages, this relationship is obscured because the translator chose to use a different equivalent instead (in this case, code-switched, but not necessarily so in other sentences).

8.4 Processing pipeline

The data provided on the “Languages Home” website clearly holds potential for the purpose of extracting “gold” Hindi-parallel lexicons for 20 languages. However, there are two primary problems to be solved to that end:

- (Content) The most obvious problem is that it consists of parallel sentences, not bilingual lexicons; we must extract the latter from the former.
- (Script) The data (both the Indic source and English target) are in the Roman script; further, in a non-standard “casual” transliteration for Indic languages.

Of course, we also have the general problems of noise, as described above, as well as lossiness arising from the nature of the casual translations as opposed to the ideal comprehensiveness and accuracy of a bilingual lexicon.

We pass the above data (consisting of 21 individual English-parallel files, one per language), through the pipeline as shown in Figure 8.1.

Each step of the process is explained below:

[1]By itself, this difference is not a bad thing given that the purpose of this website is language learning. In Hindi, the given parallel sentence is absolutely natural-sounding - people do often code-switch the word “pen”. Code-switching with English may be less common in less urban languages such as Bagheli; thus accounting for the use of the native word “kalam”.

1. **Getting the data** : The data per language was manually downloaded from the website. This data is simply a list of sentences, alternating between source and target.

2. **Formatting and Cleaning** : The above data was formatted into the following structure : *(Indic) source < sep > (English) target*, the required input format of further steps. The data also underwent some cleaning, i.e. removal of some words manually identified as not part of the translation (such as "(respect)", "(he)"/"(she)" and so on. Other standard preprocessing was also applied, such as removal of punctuation and lower-casing.

3. **Transliteration** : As mentioned, the given data are in the Roman script. For usage in our project, as well as general usage, we would of course prefer the evaluation data to be in the Devanagari script. Since we do not have the resources and human resources to perform a manual transliteration for all 21 languages, we instead use automated transliteration given by the *indic-trans* library[2] [Bhat et al., 2015]. We also attempted to use the IndicNLP Library [Kunchukuttan, 2020]; however, the former worked much better for the Roman spellings we are dealing with; the latter expects certain transliteration schemes and performs poorly on our data.

4. **Parallelizing with Hindi** : In this step, we re-parallelize all languages with Hindi instead of English; this is simple given that we do have the Hindi equivalents of the English sentences and, as mentioned, the English sentences remain more-or-less identical over all languages. That is, we obtain the structure *(Hindi) source < sep > (Indic) target*. The reason for this step is because we expect that it will be easier and more accurate to word align Indic languages with Hindi sentences as compared to English sentences; Indic languages may share syntactic properties with Hindi, including particles, function words, inflection types, etc. that English does not have, and that will help the alignment algorithm to find equivalents not only for content words such as nouns but also inflected forms, syntactic function words, etc. Although the majority of the English sentences are identical across languages, we note data loss of a few sentences per language in this step for mismatch of the English targets in the Hindi set as compared to another language set.

5. **Aligning against Hindi** : We use the FAST_ALIGN algorithm [Dyer et al., 2013] to extract word-alignments over the given Hindi-parallel data. We do this with Hindi as source as well as target; the resulting lexicons may have different use-cases. In this work, we mainly use the the lexicons with Hindi as source as the other 20 languages as targets, to allow easy extrapolation of equivalents for any other language pair.

6. **Aligning against English** : We do the same for the English-parallel data, with English as target. However, the quality of these alignments is not so good, as expected, and we do not use these lexicons in this work.

[2] `https://github.com/libindic/indic-trans`

```
{
  "उसने": {
    "उ": 8
  },
  "पैसा": {
    "पैसा": 1,
    "रुपिया": 1
  },
  "क्यों": {
    "कहें": 7,
    "कहे": 1
  },
  ...
}
```


Figure 8.2: Extract from the Hindi-Awadhi extracted lexicon. The counts shown are the number of times the target key was aligned with the source over all parallel sentences. This number can be useful to filter out noisy alignments.

7. **Reading alignments** : Finally, it remains to read the resulting alignments and construct a bilingual lexicon. We structure the lexicon as a JSON, shown in Figure 8.2, allowing a source key with many possible target values. Each target value is marked with the number of times it was aligned with the source key in question.

The above pipeline can be run fully automatically, given the path to the raw data and the path to the cloned FAST_ALIGN directory. An additional step we attempted to incorporate was in the transliteration stage; hoping to correct wrong transliterations, we performed the following for each transliterated word:

- Check whether transliterated word exists in collected corpus
- If not, choose the closest word from the corpus by a variant of the normalized edit distance measure.

We tried a number of variants for this purpose, with phonological motivations. For example, a variant only allowing vowel changes, a variant disallowing changes near the beginning of the word, and even a variant only allowing changes to the (Devanagari) character representing /a/, which is often erroneously added in transliterations, because of the overloading of the Roman character “a” with both /a/ and /ə/, differently represented in Devanagari. However, even in the most restrictive variants, we find that the results of this process are too noisy and error-prone; we decided to remove it from the pipeline.

The above pipeline is available here: `https://github.com/niyatibafna/north-indian-dialect-modelling/tree/main/evaluation_languages_home`.

Language	Total in corpus	Unique in corpus	Total in test	Unique in test	Common	Frac covered in corpus[1]	Frac covered in test[2]
brajbhasha	156986	30194	613	166	97	0.13	0.65
angika	1253545	91757	691	180	111	0.10	0.60
maithili	218491	41434	627	162	89	0.09	0.54
magahi	79405	16942	667	174	82	0.11	0.65
hindi-urdu	7100394	197355	673	172	165	0.25	0.98
awadhi	490877	53103	603	154	116	0.05	0.82
rajasthani	187708	34360	691	174	131	0.12	0.83
hariyanvi	232526	27431	611	159	125	0.14	0.85
bhil	27246	5557	649	179	69	0.12	0.49
chattisgarhi	83073	14463	591	142	98	0.16	0.75
nepali	688865	104687	517	146	79	0.05	0.61
bajjika	7412	2788	663	151	55	0.13	0.53
koraku	15508	2278	535	135	18	0.04	0.22
malwi	9626	2883	669	169	51	0.12	0.45
sindhi	52659	11850	597	165	60	0.09	0.50
bhojpuri	196513	34051	679	163	120	0.17	0.82
garwali	90234	22655	621	176	91	0.07	0.63
marathi	3109	1685	495	142	31	0.05	0.36
kumaoni	1013	441	557	186	17	0.10	0.15
bundeli	26902	7991	623	167	89	0.12	0.62

Table 8.3: Evaluation data statistics post-transliteration. [1] This reports the fraction of the corpus (token-wise) that is contained in the test, vice-versa for [2].

8.5 Collected data

8.5.1 General statistics

The resulting bilingual lexicons may contain a fair amount of noise, mainly due to the transliteration and word alignment steps. Most importantly, many of the word correspondences, even if perfectly aligned and transliterated, are not in fact cognates or borrowings, but simply other lexical equivalents. Although we would like to ideally perform a post-editing of these lexicons, with native speakers correcting spellings as well as deleting false word equivalents and adding correct ones, this is infeasible at the moment.

We would therefore like to gauge the quality of the evaluation lexicons across two dimensions i.e. firstly, the quality of the transliteration into Devanagari, and secondly, the word-alignments themselves.

We perform a manual evaluation of the transliteration for two languages: Hindi and Marathi, and evaluation of the Hindi-paired word alignments for a single language i.e. Hindi-Marathi.[3]

We report general statistics of the English-parallel transliterated data in Table 8.3, and the equivalent table for the Hindi-parallel data in Table 8.4. The former is only provided here to observe data loss due to imperfect parallellization with Hindi because of variations in the English sentences provided over all languages; for all further use, we refer to Table 8.4.

8.5.2 Quality testing on Marathi

As mentioned before, we conduct a manual evaluation of the Hindi/Urdu-Marathi bilingual lexicon. The transliteration was evaluated by simply calculating the percentage of Marathi words in the lexicon were (entirely) accurately transliterated. This percentage is low - only 54.6%, or 71 out of a total of 130 words, were correctly transliterated.

The word alignments were evaluated in the Hindi (source), Marathi (target) direction. A single alignment was marked as correct if any one of the listed targets was accurate. A word in the collected Marathi bilingual lexicon has on average 1.2 targets, and a maximum of 3. We found that 100 of 136, or 73.5% of word-alignments by this measure are correct. Note that we do not consider whether the targets are accurately transliterated while evaluating the word alignments.

8.6 Conclusion

Collecting data for evaluation is difficult given the extreme scarcity of basic resources for Band 3 languages. The bilingual lexicons that we have built may be useful as a first step, if human resources for post-editing the transliterations and alignments become available.

For the purpose of this project, we will continue to use these lexicons despite clear problems of noise as evident in the manual examination; however, we will take the lexicons as a relative rather than absolute indicator of performance, supplemented with qualitative analysis.

[3]This is the only language that I speak and am qualified to evaluate, besides Hindi.

Language	Total in corpus	Unique in corpus	Total in test	Unique in test	Common in corpus and test	Frac covered in corpus[1]	Frac covered in test[2]
brajbhasha	156986	30194	299	161	93	0.12	0.65
angika	1253545	91757	310	165	102	0.09	0.60
maithili	218491	41434	273	147	81	0.09	0.54
magahi	79405	16942	326	172	81	0.11	0.64
hindi-urdu	7100394	197355	336	171	165	0.25	0.98
awadhi	490877	53103	281	145	109	0.05	0.82
rajasthani	187708	34360	312	161	124	0.11	0.84
hariyanvi	232526	27431	298	156	123	0.13	0.86
bhil	27246	5557	319	177	68	0.12	0.48
chattisgarhi	83073	14463	267	134	95	0.16	0.76
nepali	688865	104687	203	118	65	0.04	0.62
bajjika	7412	2788	317	149	55	0.13	0.53
koraku	15508	2278	262	132	17	0.04	0.23
malwi	9626	2883	325	163	51	0.12	0.46
sindhi	52659	11850	250	141	55	0.09	0.51
bhojpuri	196513	34051	303	146	110	0.16	0.83
garwali	90234	22655	275	161	86	0.07	0.64
marathi	3109	1685	230	130	29	0.05	0.37
kumaoni	1013	441	250	171	16	0.10	0.16
bundeli	26902	7991	272	147	82	0.12	0.63

Table 8.4: Evaluation data statistics post-transliteration, after aligning with Hindi. [1] This reports the fraction of the corpus (token-wise) that is contained in the test, vice-versa for [2].

Chapter 9

Results and discussion

9.1 Quantitative results

Our main results are presented in Table 9.1. We report $Prec@K$ over the bilingual lexicons presented in Chapter 8. A precision point is calculated per source word such that any of the top K predicted targets exist in the evaluation target set. (Specifically, we report $Prec@5$.) Note that recall is the same for all approaches. This is because we used the same set of source Hindi words for the final lexicon for all approaches; therefore, recall (defined as the coverage of the predicted lexicon over the evaluation lexicon) is the same regardless of approach. This allows us to compare precision directly.

There is no clear quantitative winner; SEM_JW performs slightly better than the other approaches on average.

We note that general performance seems low over all approaches. Cognate identification methods usually work at a much higher accuracy [Beinborn et al., 2013, Fourrier et al., 2021] in the range of 70-90%. The low accuracies that we record are due to a number of factors: much lower resource range, lack of aligned word lists, lemmatizers, or supervision, as well as noise in the evaluation data itself.

9.2 Qualitative analysis: general overview

We take a high level look at the patterns in the outputs of each of the different approaches.

9.2.1 NED/JW

The NED/JW approaches are the most transparent. They are often able to capture the correct answer, especially for longer words, because the closest candidate in edit distance is likely to be the right answer, or if even if not exactly correct, somewhere in the ballpark semantically and morphologically. However, the outputs of these approaches are of course uninformed by semantics, and we often get outputs (perhaps the second or third prediction) that are entirely off, as can be expected from theoretically unmotivated character substitutions. The JW metric has an edge over NED when it comes to quality of outputs (not always reflected in the quantitative measure).

	Total	Found	NED	JW	EMT	SEM_JW	SEM_EMT
angika	141.0	116.0	21.6	20.7	21.6	**22.4**	21.6
awadhi	148.0	123.0	**28.5**	26.8	22.0	26.0	25.2
bajjika	149.0	123.0	13.8	**15.4**	13.8	14.6	11.4
bhil	156.0	128.0	19.5	**21.1**	17.2	18.8	18.0
bhojpuri	139.0	115.0	31.3	28.7	**32.2**	30.4	29.6
brajbhasha	155.0	127.0	33.9	**34.6**	32.3	33.9	32.3
bundeli	139.0	117.0	26.5	25.6	25.6	**30.8**	26.5
chattisgarhi	136.0	115.0	25.2	26.1	24.3	**28.7**	26.1
garwali	143.0	120.0	**15.8**	**15.8**	15.0	**15.8**	14.2
hariyanvi	153.0	126.0	38.1	41.3	37.3	**43.7**	42.9
koraku	140.0	116.0	**1.7**	0.9	**1.7**	**1.7**	0.9
kumaoni	138.0	118.0	**5.1**	4.2	**5.1**	**5.1**	4.2
magahi	159.0	129.0	17.8	**20.9**	18.6	**20.9**	17.1
maithili	140.0	117.0	17.9	17.1	16.2	18.8	**20.5**
malwi	153.0	125.0	**24.8**	22.4	20.0	20.0	15.2
marathi	138.0	116.0	**7.8**	5.2	4.3	1.7	3.4
nepali	105.0	95.0	**12.6**	**12.6**	9.5	9.5	7.4
rajasthani	144.0	120.0	30.8	29.2	27.5	**31.7**	30.0
sindhi	134.0	114.0	10.5	**13.2**	7.9	10.5	9.6
Avg.	142.6	118.9	20.1	20.2	18.5	**20.3**	18.7

Table 9.1: Prec@5 for all languages, for cognate induction.

9.2.2 EMT

The EMT approach was intended to remedy the above problem, by weighting the edit distance matrix differentially based on learnt weights. While this seems like the logical next step, the approach itself does a little worse than the above approaches. We attribute this to a bad seed; this approach basically depends on the seed obtained from simple NED to get started, and if it meanders down a mistaken path, that error tends to magnify itself due to the iterative nature of the algorithm. Further, the approach needs much more attention to priors; what should those initial transform probabilities really be? How much probability mass should we give the self-transform?

Taking a look into the learnt probabilities, we see that it learns sometimes expected relationships e.g. the relationship between /i/ and /iː/, shifts between other vowels, or the fact that some rarely used characters are likely to be deleted. However, while this is a step in the right direction, the approach is not able to convert this potential into good final outputs; further, magnified errors often result in even worse final outputs than simple NED/JW.

9.2.3 SEM_*

The SEM_* approaches are intended to address the fundamental inadequacy in the above approaches: the fact that they do not exploit the shared semantics of cognates. SEM_JW is accordingly better at producing outputs that are semantically related, besides the required cognates. Top predictions tend to be similar

to those of NED/JW, but SEM_JW produces a better collection of outputs from the perspective of bilingual lexicons, especially since it is less biased against a higher number of substitutions. That is to say, if we have two solutions, each containing K predicted targets, that both lack the accurate target prediction, we prefer to have a cluster of semantically related target words (as produced by SEM_JW) in a bilingual lexicon rather than words with similar spellings (as produced by NED/JW); similarly, even if both solutions do include a correct target, we would like for the other predictions to be semantically rather than orthographically related to the source word. However, for many words, the method produces rather Hindi-like outputs, probably as a result of the persisting problem of language-wise clustering in the spaces.[1] SEM_EMT still suffers from the same problems as before; we see therefore that a stronger orthographic distance metric such as JW is better able to spot the cognate if any from semantically related words.

9.3 Qualitative analysis: different aspects

We analyse the performance of the baselines, EMT, and the SEM_* approaches based on how well they capture the different facets of lexical equivalents with a shared origin (i.e. cognates with a shared genealogy or borrowings with a shared origin lexeme).

9.3.1 Variant inflectional endings

Learning the different inflectional endings of word classes is a crucial task when it comes to cognate identification across dialectical variations/closely related languages. That is, we want the approach to learn the correspondences between inflections in the two dialects.

This task is not performed well by any of the current approaches, although it is specifically what *EMT is targeted at. However, *EMT faces the problems discussed above.

In terms of being able to produce the right answer, we see an intuitive split between common and rare words when it comes to other approaches. For common words, SEM_JW is likely to perform better than the other approaches because the word is likely to be well embedded; the right inflection is likely to be near by in the semantic space, and subsequently selected by the JW metric. In these cases, especially for shorter words, NED/JW are likely to be derailed by irrelevant words. We see an example of this situation in with an extremely common verb "said", only three characters long. See Figure 9.4 for the outputs of the various approaches. However, for rare words that are badly embedded, the SEM_* may have irrelevant nearest neighbours, producing incorrect output.

9.3.2 Correct semantics

This refers to getting the general semantics of predictions right, even if the predicted words are not cognates. Naturally, this is performed best by the SEM_*

[1] This problem may be mitigated with a higher target frequency threshold.

	NED	JW	EMT	SEM_JW	SEM_EMT	Gold
1	यात्रा	यात्रा	यात्रा	यात्रा	यात्रा	सफर
2	मात्रा	यात्राएँ	मात्रा	यात्राएँ	यात्री	-
3	यात्री	यात्री	यात्री	यात्रायें	यात्राएँ	-
4	यात्राएँ	यात्रायें	तूरा	यात्री	यात्रायें	-
5	यंत्रणा	मात्रा	-	मात्रा	यात्रियों	-

Figure 9.1: Hindi source word: /yaːt̪raː/ (journey). NED gives /maːt̪raː/ (measure) as second guess and /yənt̪rəɳaː/ (strategy) as fifth prediction (unrelated but similar in spelling). JW also predicts /maːt̪raː/ but as its fifth option. The predictions of the SEM_* approaches are various inflectal/derivational forms of /yaːt̪raː/ including "travellers". The gold in this case is a non-cognate of the source word in Hindi; this is one of the cases where a cognate does exist but is not represented by the evaluation data.

approaches, although as discussed before, the NED/JW approaches do better than expected. For examples, see Figures 9.2, 9.1, and 9.3.

9.3.3 Sound changes

Sound change is one of the fundamental phenomenons of cognacy, and can be understand in the case of borrowing in the sense of changed pronunciations. Unfortunately, we do not have the theoretical data of attested sound changes across these dialects in order to be best able to check which approach performs best in this respect.

In general, we face issues of noisy evaluation data as well as scarce theoretical data, either for good seeds and for evaluation. The SEM_JW produces overall the most respectable outputs as such, although this is more true for common words.

	NED	JW	EMT	SEM_JW	SEM_EMT	Gold
1	रास्ते	रास्ते	रास्ते	रास्ते	रास्ते	से
2	रास्ता	रस्ते	रास्ता	रास्ता	रास्ता	-
3	रस्ते	रास्ता	वास्ते	रास्तों	रस्ते	-
4	रिस्ते	रिस्ते	रस्ते	रस्ते	रास्तों	-
5	वास्ते	राते	-	वास्ते	राहों	-

Figure 9.2: Source word: /rəstaː/ (road/way). While no approach produces the gold prediction, SEM_* is able to produce semantically relevant words, including a synonym /rahõ/ (roads) whereas NED/JW produce orthographically similar but unrelated words such as /rate/ (nights) and /riste/ (relationships).

	NED	JW	EMT	SEM_JW	SEM_EMT	Gold
1	मीठा	मीठा	मीठा	मीठा	मीठा	मीठ्
2	मीठे	मीठ	मीरा	मीठ	मीठे	-
3	मीठ	मठवा	मीना	मीठे	म	-
4	मीरा	मीठे	मीता	मीठी	मीठरस	-
5	मीना	मीरा	-	माठा	खट्टा	-

Figure 9.3: Hindi source word: /miːt^{h}aː/ (sweet). NED produces last two predictions /miːra/ and /miːna/ (both names of women, unrelated to word). SEM_EMT produces more semantically coherent (if not cognate) predictions, including /miːt^{h}rəs/ (sweet juice) and /k^{h}ətaː/ (sour).

	NED	JW	EMT	SEM_JW	SEM_EMT	Gold
1	कहा	कहा	कहा	कहा	कहा	कहलाह
2	कहना	कहना	कहना	कहात	क	कहल
3	कहाँ	कहाँ	कहाँ	कहाई	कहौं	-
4	एकहा	कहमा	एकहा	कहनाम	लजाते	-
5	कहमा	कहां	-	कह	पूछा	-

Figure 9.4: Hindi source word: /kəhaː/ (said). SEM_JW approach performs the best, resulting in Bhojpuri equivalents (except the third prediction) and inflections. SEM_EMT also results in semantically correct outputs (for all but the fourth prediction). The NED/JW approaches produce orthographically close words that are semantically unrelated, e.g. /kəhãː/ (where).

Conclusion and Future Work

Many North Indian (Indic) languages are entirely lacking resources as well as attention in NLP: in this work, we consider the "Hindi Belt" dialect continuum considered to contain several dozens of languages and dialects. The languages are sociopolitically disadvantaged, many of them without official status or recognition; however, their native speaker count runs up to tens of millions.

Research in these languages has the potential to provide support to a large speaker base, as well as provide insight into linguistic aspects dialect continua and computational insights into building tools for closely related languages.

In this work, we work with 26 languages of this Indic dialect continuum associated with the Hindi Belt, although we include certain languages not included under this term, and we do not cover all languages that may be considered part of it. Our goal is to contribute basic resources such as monolingual data and bilingual lexicons, conduct preliminary experiments to probe cross-lingual relationships in the dialect continuum, as well as to explore algorithms that may be used to exploit the shared lineage of these languages in a relatively fundamental task of induction of cognate/borrowing lexicons - i.e. finding word pairs across two languages that are either descended from the same root or are borrowed from one language into the other (or from a third language altogether).

This work was conducted in four stages:

1. Data Collection: Collecting monolingual corpora
2. Data Probing: Looking into the collected data at the character, subword, and lexical level to find relationships between different languages and subsets of languages.
3. Evaluation Data Collection: Collecting "gold" lexicons for evaluation of cognate induction methods.
4. Cognate induction: Experimenting with different methods to induce bilingual cognate lexicons for each language with Hindi-Urdu.

We summarize our contributions and findings in each of these below, along with outlining work as yet to be done.

9.4 Data Collection and Probing

We crawled monolingual corpora for as many languages of the Indic dialect continuum under consideration as we could find. We created a collection of folksongs and poetry in 26 languages, forming the largest collection of data from closely

related dialects in the number of languages as far we are aware. Of these, 16 languages (not including Khadi Boli) had no preexisting data available systematically to the NLP community. We were authorized to make the folksongs in all languages available, but not the poetry, due to copyright reasons. However, our crawler is publicly available and may be used to crawl the website under question.

We conducted experiments to gauge pairwise cross-lingual similarities in the given set of languages, using overlap-based as well as KL Divergence-based metrics at the character, subword and lexical level. These experiments confirmed some expectations from prior genealogical knowledge about the languages.

The limitations of the dataset include that it has very little data (just few hundreds of tokens) for 3 languages. The folksongs contained in the dataset are difficult to date; therefore, parts of the data may not be representative of the language as it is spoken today. With Marathi, which has a rich literary tradition, we also see that its poetry dates back several centuries to an almost unrecognizable Marathi, although this is rare in the dataset in general.

One of the most crucial tasks remaining to be performed in this regard is the development of a good quality language identification tool. Our labelled dataset can be used for this purpose, and the resulting tool can be then be used for collecting more data from the web.

9.5 Collecting Evaluation Data

We collect evaluation data for the task of cognate induction for 20 languages of our collection against Hindi-Urdu. Reviewing a wide range of available resources such as blogs, language learning websites, and others, we choose the website with the best coverage of languages as well as a decent (overlapping) source vocabulary that each of the languages can be mapped to.

This resource, consisting of artificially simply English phrases and their translations into target languages (all in casually transliterated Roman script), required considerable post-processing and adaptation to make it usable for our purpose. While we are prohibited from publishing derivative data from this website, our processing pipeline is publicly available to run on raw data from the website.

This collection of evaluation lexicons has many problems of noise, incompleteness, and transliteration, resulting from errors at different points in the automatic pipeline. We would like to have post-processing and editing of each lexicon by native speakers of the respective languages, involving deletion of incorrect options, spelling correction, and possibly adding cognate equivalents where they exist and are not listed.

9.6 Cognate Induction

We explore methods of cognate/borrowing induction in a bilingual setup, with source words from Hindi and target cognates from each of the languages under consideration. Our baselines consist of optimizing simply orthographic metrics such as normalized edit distance (NED) and the Jaro-Winkler metric (JW).

We try two new methods in order to exploit the two primary aspects of cognacy, i.e. shared orthographic features as well as semantic relatedness. In the first (called EMT), we implement an expectation-maximization approach in order to learn sound changes for a given pair of languages. In the second approach (called SEM_JW), we use bilingual embedding spaces to narrow down possible candidates that are then judged by orthographic features. Finally, we also combine the two approaches (SEM_EMT).

We find that the SEM_JW does slightly better than the other approaches. The current implementation of the *EMT approach suffers from a number of problems, including bad prior initialization. Although it has potential in being theoretically most capable of learning regular patterns of sound change between two languages, it struggles to overcome its bad initialization as well as lack of a good seed, since its initial datapoints are selected by edit distance. We find that the bottleneck in the SEM_JW approach is (naturally) the quality of the learnt embeddings; this tasks faces the obstacles of a large data disparity between the two languages, as well as general data scarcity in the target language.

Future work can address the above issues; i.e. honing the EMT approach, as well as investigating better bilingual embeddings, or experiments with multilingual embeddings for the full set of languages. The primary inadequacy of all these approaches is their inability to (explicitly) capture language-pair specific correspondences. An extension of this work could focus on refining something akin to the SEM_EMT. Improvements could include searching the hyperparameter space for better priors, as well as investigating a better bi/multilingual space. Further, the approach could try to treat words differently based on frequency and an initial set of outputs.

Bibliography

Marc Allassonnière-Tang and Michael Dunn. The evolutionary trends of grammatical gender in Indo-Aryan languages. *Language Dynamics and Change*, 11 (2):211–240, 2020.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 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 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1073. URL `https://aclanthology.org/P18-1073`.

Duygu Ataman, Wilker Aziz, and Alexandra Birch. A Latent Morphology Model for Open-Vocabulary Neural Machine Translation. *arXiv e-prints*, pages arXiv–1910, 2019.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Laith H Baniata, Seyoung Park, and Seong-Bae Park. A neural machine translation model for arabic dialects that utilizes multitask learning (mtl). *Computational intelligence and neuroscience*, 2018, 2018.

Khuyagbaatar Batsuren, Gábor Bella, and Fausto Giunchiglia. Cognet: A large-scale cognate database. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 3136–3145, 2019.

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. Cognate production using character-based machine translation. In *Proceedings of the sixth international joint conference on natural language processing*, pages 883–891, 2013.

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. IIIT-H System Submission for FIRE2014 Shared Task on Transliterated Search. In *Proceedings of the Forum for Information Retrieval Evaluation*, FIRE '14, pages 48–53, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3755-7. doi: 10.1145/2824864.2824872. URL `http://doi.acm.org/10.1145/2824864.2824872`.

Mikhail Bilenko and Raymond J Mooney. Employing Trainable String Similarity Metrics for Information Integration. In *IIWeb*, pages 67–72, 2003.

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

Ondřej Bojar, Vojtěch Diatka, Pavel Rychlý, Pavel Straňák, Vít Suchomel, Aleš Tamchyna, and Daniel Zeman. HindMonoCorp 0.5, 2014. URL `http://hdl.handle.net/11858/00-097C-0000-0023-6260-A`. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Hailong Cao and Tiejun Zhao. Word embedding transformation for robust unsupervised bilingual lexicon induction. *arXiv preprint arXiv:2105.12297*, 2021.

Chundra Cathcart. Toward a deep dialectological representation of Indo-Aryan. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 110–119, 2019.

Chundra Cathcart and Taraka Rama. Disentangling dialects: a neural approach to indo-aryan historical phonology and subgrouping. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 620–630, 2020.

Çağrı Çöltekin. Cross-lingual morphological inflection with explicit alignment. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 71–79, 2019.

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

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

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

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

Tejas Indulal Dhamecha, Rudra Murthy V, Samarth Bharadwaj, Karthik Sankaranarayanan, and Pushpak Bhattacharyya. Role of Language Relatedness in Multilingual Fine-tuning of Language Models: A Case Study in Indo-Aryan Languages. *arXiv preprint arXiv:2109.10534*, 2021.

Zi-Yi Dou, Zhi-Hao Zhou, and Shujian Huang. Unsupervised bilingual lexicon induction via latent variable models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 621–626, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1062. URL `https://aclanthology.org/D18-1062`.

Pankaj Dwivedi and Somdev Kar. Sociolinguistics and phonology of Kanauji. In *International Conference on Hindi Studies*, 2016.

Chris Dyer, Victor Chahuneau, and Noah A Smith. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, 2013.

M.A. Egorova and A.A. Egorov. The ancestral homeland of the carriers of the Proto-Indo-European language: Mathematical models for the study of linguistic information. *Automatic Documentation and Mathematical Linguistics*, 53 (3):127–137, 2019.

Wael Farhan, Bashar Talafha, Analle Abuammar, Ruba Jaikat, Mahmoud Al-Ayyoub, Ahmad Bisher Tarakji, and Anas Toma. Unsupervised dialectal Neural Machine Translation. *Information Processing & Management*, 57(3):102181, 2020.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. Morphological Inflection Generation Using Character Sequence to Sequence Learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 634–643, 2016.

Clémentine Fourrier, Rachel Bawden, and Benoît Sagot. Can Cognate Prediction Be Modelled as a Low-Resource Machine Translation Task? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Bangkok, Thailand, August 2021. URL `https://hal.inria.fr/hal-03243380`.

Alexandre François. Trees, waves and linkages. *The Routledge handbook of historical linguistics*, pages 161–189, 2015.

Andrew Garre. Convergence in the formation of indo-european subgroups: Phylogeny and chronology. *Phylogenetic methods and the prehistory of languages. Cambridge: McDonald Institute for Archaeological Research*, pages 139–151, 2006.

Andrew Garrett, Nadine M Tang, and Bruce L Smith. New perspectives on Indo-European phylogeny and chronology. *Proceedings of the American Philosophical Society*, 162(1):25–38, 2018.

Andrej Gisbrecht, Alexander Schulz, and Barbara Hammer. Parametric nonlinear dimensionality reduction using kernel t-sne. *Neurocomputing*, 147:71–82, 2015.

Fausto Giunchiglia, Khuyagbaatar Batsuren, and Abed Alhakim Freihat. One world–seven thousand languages. In *Proceedings 19th international conference on computational linguistics and intelligent text processing, CiCling2018*, pages 18–24, 2018.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, 2012.

Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. FRAGE: Frequency-agnostic word representation. *Advances in neural information processing systems*, 31, 2018.

Imane Guellil, Faical Azouaou, and Mourad Abbas. Neural vs statistical translation of Algerian Arabic dialect written with Arabizi and Arabic letters. In *The 31st pacific asia conference on language, information and computation paclic*, volume 31, page 2017, 2017.

David Hall and Dan Klein. Finding cognate groups using phylogenies. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1030–1039. Citeseer, 2010.

Bradley Hauer and Grzegorz Kondrak. Clustering semantically equivalent words into cognate sets in multilingual lists. In *Proceedings of 5th international joint conference on natural language processing*, pages 865–873, 2011.

Michael A Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, 2021.

Wilbert Heeringa and John Nerbonne. Dialect areas and dialect continua. *Language Variation and Change*, 13(3):375–400, 2001.

Péter Jeszenszky and Robert Weibel. Measuring boundaries in the dialect continuum, 2015.

Saurav Jha, Akhilesh Sudhakar, and Anil Kumar Singh. Learning Cross-lingual Phonological and Orthagraphic Adaptations: A Case Study in Improving Neural Machine Translation between Low-Resource Languages. *arXiv preprint arXiv:1811.08816*, 2018.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. The State and Fate of Linguistic Diversity and Inclusion in the NLP World. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, 2020.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.445. URL `https://aclanthology.org/2020.findings-emnlp.445`.

Diptesh Kanojia, Kevin Patel, Malhar Kulkarni, Pushpak Bhattacharyya, and Gholemreza Haffari. Utilizing Wordnets for Cognate Detection among Indian Languages. In *Proceedings of the 10th Global Wordnet Conference*, pages 404–412, 2019.

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

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180, 2007.

Grzegorz Kondrak. A new algorithm for the alignment of phonetic sequences. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*, 2000.

Amit Kumar, Rajesh Kumar Mundotiya, and Anil Kumar Singh. Unsupervised approach for zero-shot experiments: Bhojpuri–Hindi and Magahi–Hindi@LoResMT 2020. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*, pages 43–46, Suzhou, China, December 2020. Association for Computational Linguistics. URL `https://aclanthology.org/2020.loresmt-1.6`.

Ritesh Kumar, Bornini Lahiri, Deepak Alok, Atul Kr Ojha, Mayank Jain, Abdul Basit, and Yogesh Dawer. Automatic Identification of Closely-related Indian Languages: Resources and Experiments. *arXiv preprint arXiv:1803.09405*, 2018.

Anoop Kunchukuttan. The IndicNLP Library. `https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf`, 2020.

Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, Avik Bhattacharyya, Mitesh M Khapra, Pratyush Kumar, et al. AI4bharat-indicnlp corpus: Monolingual corpora and Word Embeddings for Indic languages. *arXiv preprint arXiv:2005.00085*, 2020.

Surafel M Lakew, Aliia Erofeeva, and Marcello Federico. Neural Machine Translation into Language Varieties. *arXiv preprint arXiv:1811.01064*, 2018.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*, 2017.

Yu-Chun Li, Hua-Wei Wang, Jiao-Yang Tian, Rui-Lei Li, Zia Ur Rahman, and Qing-Peng Kong. Cultural diffusion of indo-aryan languages into bangladesh: A perspective from mitochondrial dna. *Mitochondrion*, 38:23–30, 2018.

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

Johann-Mattis List. Sequence comparison in historical linguistics. In *Sequence Comparison in Historical Linguistics*. Düsseldorf University Press, 2014.

Johann-Mattis List. Automatic inference of sound correspondence patterns across multiple languages. *Computational Linguistics*, 45(1):137–161, 2019.

Pulkit Madaan and Fatiha Sadat. Multilingual neural machine translation involving indian languages. In *Proceedings of the WILDRE5–5th Workshop on Indian Language Data: Resources and Evaluation*, pages 29–32, 2020.

Karima Meftouh, Salima Harrat, Salma Jamoussi, Mourad Abbas, and Kamel Smaili. Machine translation experiments on padic: A parallel arabic dialect corpus. In *Proceedings of the 29th Pacific Asia conference on language, information and computation*, pages 26–34, 2015.

Diwakar Mishra and Kalika Bali. Hindi dialects phonological transfer rules for verb root. In *13th oriental COCOSDA-2010 conference in coordination with International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques held at Kathmandu. Nepal*, pages 24–25, 2010.

Diwakar Mishra and Kalika Bali. A Comparative Phonological Study of the Dialects of Hindi. In *ICPhS*, volume 17, pages 17–21, 2011.

Rajesh Kumar Mundotiya, Shantanu Kumar, Umesh Chandra Chaudhary, Supriya Chauhan, Swasti Mishra, Praveen Gatla, Anil Kumar Singh, et al. Development of a Dataset and a Deep Learning Baseline Named Entity Recognizer for Three Low Resource Languages: Bhojpuri, Maithili and Magahi. *arXiv preprint arXiv:2009.06451*, 2020.

Rajesh Kumar Mundotiya, Manish Kumar Singh, Rahul Kapur, Swasti Mishra, and Anil Kumar Singh. Linguistic Resources for Bhojpuri, Magahi, and Maithili: Statistics about Them, Their Similarity Estimates, and Baselines for Three Applications. *Transactions on Asian and Low-Resource Language Information Processing*, 20(6):1–37, 2021.

Garrett Nicolai, Saeed Najafi, and Grzegorz Kondrak. String transduction with target language models and insertion handling. *SIGMORPHON 2018*, page 43, 2018.

Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51, 2003.

Atul Kr Ojha. English-Bhojpuri SMT System: Insights from the Karaka Model. *arXiv preprint arXiv:1905.02239*, 2019.

Atul Kr. Ojha, Valentin Malykh, Alina Karakanta, and Chao-Hong Liu. Findings of the LoResMT 2020 shared task on zero-shot for low-resource languages. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*, pages 33–37, Suzhou, China, December 2020. Association for Computational Linguistics. URL `https://aclanthology.org/2020.loresmt-1.4`.

Barun Patra, Joel Ruben Antony Moniz, Sarthak Garg, Matthew R Gormley, and Graham Neubig. Bilingual Lexicon Induction with Semi-supervision in Non-Isometric Embedding Spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 184–193, 2019.

Maxwell P Phillips. *Dialect continuum in the Bhil tribal belt: Grammatical aspects*. PhD thesis, SOAS, University of London, 2012.

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

Taraka Rama and Anil Kumar Singh. From bag of languages to family trees from noisy corpus. In *Proceedings of the International Conference RANLP-2009*, pages 355–359, 2009.

Yves Scherrer and Benoît Sagot. A language-independent and fully unsupervised approach to lexicon induction and part-of-speech tagging for closely related languages. In *Language Resources and Evaluation Conference*, 2014.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. Cross-Lingual Alignment of Contextual Word Embeddings, with Applications to Zero-shot Dependency Parsing. 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 1599–1613, 2019.

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

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. doi: 10.3115/v1/E14-2006. URL `https://aclanthology.org/E14-2006`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yu Wan, Baosong Yang, Derek F Wong, Lidia S Chao, Haihua Du, and Ben CH Ao. Unsupervised neural dialect translation with commonality and diversity modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9130–9137, 2020.

Zihan Wang, K Karthikeyan, Stephen Mayhew, and Dan Roth. Extending Multilingual BERT to Low-Resource Languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2649–2656, 2020.

Keith Williamson. Changing spaces: linguistic relationships and the dialect continuum. *Placing Middle English in Context*, 35:141–180, 2000.

Yogendra P Yadava, Andrew Hardie, Ram Raj Lohani, Bhim N Regmi, Srishtee Gurung, Amar Gurung, Tony McEnery, Jens Allwood, and Pat Hall. Construction and annotation of a corpus of contemporary Nepali. *Corpora*, 3(2): 213–225, 2008.

Kenji Yamauchi and Yugo Murawaki. Contrasting vertical and horizontal transmission of typological features. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 836–846, 2016.

Zdeněk Žabokrtský, Niyati Bafna, Jan Bodnár, Lukáš Kyjánek, Emil Svoboda, Magda Ševčíková, and Jonáš Vidra. Towards Universal Segmentations: UniSegments 1.0. In *Proceedings of the 13th International Conference on Language Resources and Evaluation (LREC'22)*, 2022.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardzic, Nikola Ljubešić, Jörg Tiedemann, et al. Language identification and morphosyntactic tagging: The second VarDial evaluation campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 1–17, 2018.

Jinpeng Zhang, Baijun Ji, Nini Xiao, Xiangyu Duan, Min Zhang, Yangbin Shi, and Weihua Luo. Combining Static Word Embeddings and Contextual Representations for Bilingual Lexicon Induction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2943–2955, 2021.

List of Figures

List of Tables

Appendix A

TSNE Plots

Figures A.1 and A.2 show TSNE visualizations for the Joint and Upsample models in the upper and lower plots respectively.

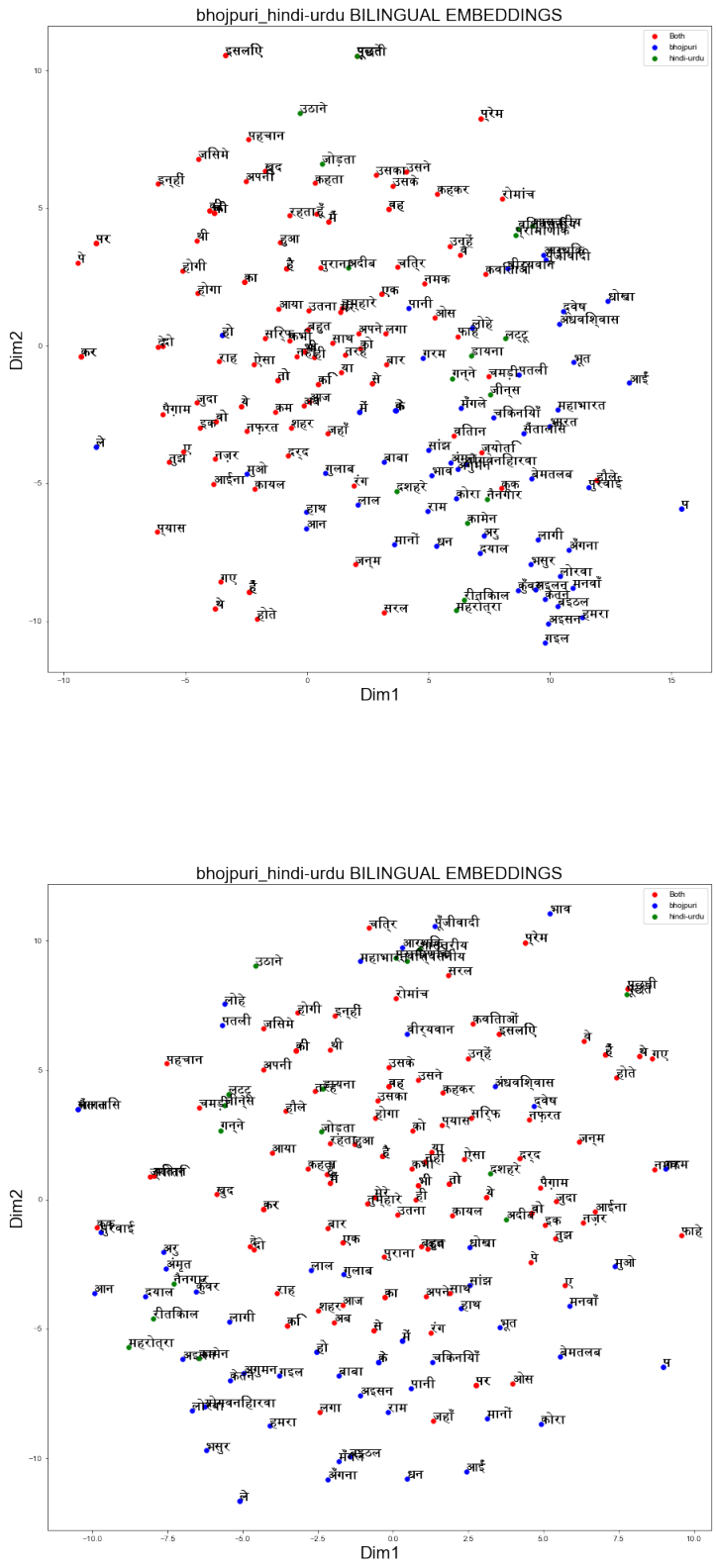

Figure A.1: Visualization of Bhojpuri-Hindi bilingual space, JOINT (up) and UPSAMPLE (down)

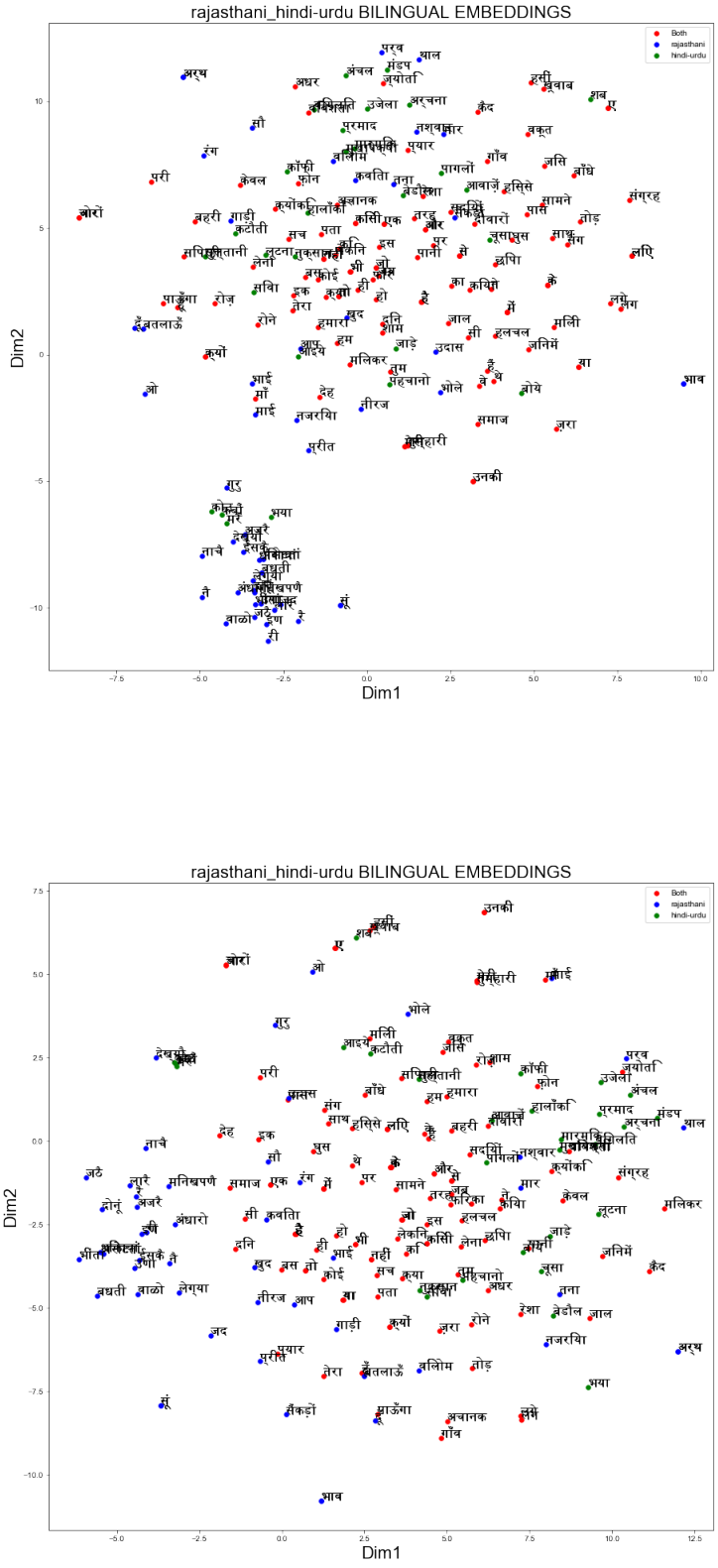

Figure A.2: Visualization of Rajasthani-Hindi bilingual space, JOINT (up) and UPSAMPLE (down)