COMPUTATIONAL METHODS FOR HISTORICAL LINGUISTICS

A Thesis
submitted to the department of Computer Science and Engineering
of International Institute of Information Technology
in partial fulfilment of the requirements
for the Masters in Technology
in Computational Linguistics

Kasicheyanula Taraka Rama

July 2009

I certify that I have read this thesis and that, in my opinion, it is fully adequate in scope and quality as a thesis for the degree of Master of Science by Research.

(Prof. B. Lakshmi Bai) Principal Adviser

Approved for the University Committee on Graduate Studies.

To my parents who spent their savings to buy me a laptop

Acknowledgements

- I would like to thank my advisor Prof. Lakshmi Bai for the support and guidance which she had given me during the work and was willing to guide me on this unconventional research topic. The animated discussions with her helped me in shaping the thesis.

- Finding the right data is always the priliminary requirement for any linguistic research. I would thank my advisor for pointing me to the right resources without which Chapter 2 and Chapter 4 would not have been written. Karthik has helped me a lot in cleaning up the data for conducting my experiments.

- I would deeply thank Sudheer Kolachina and Anil Kumar Singh for their constant encouragement and the philosophical discussions which helped shape the thesis. Their suggestions were especially helpful in writing the many parts of this thesis.

- I would like to thank my lab for the generous financial support given to me for the entire period of writing my thesis.

- The other members in the lab, Jagadeesh (who showed Machine Learning is to think like a machine) and I specifically thank him for his advise on Algorithms and Karthik for trying to prove them emprically which decisively shaped my worldview.

- Work done in Chapter 2 has been done in colloboration with Anil Kumar Singh. Work in chapter 3 has been done with both Sudheer and Anil Kumar Singh and the final chapter's work has been done in colloboration with Sudheer.

- I would like to thank Jorge Cham for drawing such an inspirational piece of drawings which showed that procastination is indeed not such a bad thing.

“God has a Book where he maintains the best proofs of all mathematical theorems, proofs that are elegant and perfect... You don’t have to believe in God, but you should believe in the Book.”

– Paul Erdös (1913-1996)

Contents

List of Tables

List of Figures

Abstract

Identification of cognates play a very important role in identifying the relationships between genetically related languages. All the attempts in this direction are based on the assumption of the existence of a proto-language in the distant past. Moreover the methods, such as comparative method, used in historical linguistics are highly dependent on human judgement and automating at least some of the steps would make the job of a historical linguist easier. I present some new methods for cognate identification as well as apply the techniques from bioinformatics, for phylogenetic tree construction, for Dravidian languages. In this process, we also propose a new system for letter to phoneme conversion.

The thesis is divided into three parts. The first part aims at identification of cognates using the phoneme feature-values. We present a set of new language independent algorithms which use distributional similarity to estimate the word similarity for identifying cognates. In the second part, we propose a new system for letter to phoneme conversion which uses algorithms from statistical machine translation and gives results comparable to the state-of-the-art systems. It takes a orthographic sequence as input and converts it into a phoneme sequence. In the third part of the thesis we show that the character based methods used in bioinformatics can be used to construct family trees in the framework of lexical diffusion. This is a novel attempt in itself and has given results which are comparable to the family trees given by the traditional comparative method.

All the above methods use the recent advances in articulatory phonetics, computational linguistics, historical linguistics and bioinformatics. These methods can not only be used for the study of language evolution but can also find use in the relevant

areas such as machine translation and transliteration. Some of the applications of the above methods have been included in the appendix. The second part can also be used for machine transliteration. Similarily the first part was used for estimating the distances between the major literary languages of Indian subcontinent and subsequently the results were used for constructing a family tree for these languages.

Chapter 1

Introduction and Background

Languages replicate themselves (and thus 'survive' from generation to generation) through a process of native-language acquisition by children. Importantly for historical linguistics, that process is tightly constrained.

- Don Ringe

Historical linguistics studies the relationships between languages as they change over time. How do we establish that two geographically distant languages such as Sanskrit and Latin are related? Providing the evidence that a pair of words from such two divergent languages are related and have indeed descended from a common word form in the past is one of the main task of historical linguistics. This problem of establishing the word similarities is excaberated for languages which donot have written records. Then the only course left is to study the modern word forms for clues to establish the **proto-form** of the **proto-language**. For this purpose, historical linguists have come up with the elegant technique called **comparative method** which constructs the proto-forms as well as establishes the language families. We discuss the basic notions and concepts used in historical linguistics in the following sections.

The research in computational linguistics draws from both computer science and linguistics and should be able to address both the audience. This thesis has two applications of computational methods to historical linguistics, namely cognate identification and phylogenetic tree construction. So we discuss the concepts in historical linguistics in detail relating to each application separately under their respective headings. Section 1.1 gives the basics of cognate identification and its relevance in the other areas of natural language processing. Section 1.2 describes the basics and the need for a letter to phoneme conversion which is very useful for cognate identification. Section 1.3 discusses briefly the ideas behind language evolution and the different frameworks in the study of language change. It also discusses now outdated method called glottochronology which was used to estimate the language divergence time based on the amount of *relatedness* between languages. The contributions of this thesis are summarised in section 1.4. An outline of the thesis is given in section 1.5.

1.1 Cognate Identification

Cognates are words of the same origin that belong to different languages. For example, the English word *beaver* and the German word *biber* are cognates descending from Proto-Germanic **bebru* and Proto-Indo-European **bher*. Identification of cognates is a major task in Historical Linguistics for constructing the family tree of the related languages. The number of cognates between two related languages decreases with time. Recurrent sound correspondences which are produced by regular sound changes are very important cues in indentifying the cognates and the reconstruction of the proto-language. These sound correspondences help in distinguishing between cognates from chance resemblances. For example, the /d/:/t/ is a regular sound correspondence between Latin and English (*ten/decem and tooth/dentem*) and helps us to identify that latin *die* 'day' is not cognate with English *day*[1]. The current research has only been able to study only a few language families. Any tool which automates this process and provides reliable results can be useful for studying new language families. Atleast the initial results provided by a tool can be used as a starting point for the study of new languages.

[1] The example has been taken from Kondrak [47]

As cognates usually have similar phonetic (and mostly orthographic) forms as well as similar meaning, string similarities can be used as the first step for identifying them. Not only orthographic similarities but also phonetic based similarities can also be used for determining the cognateness of a word pair depending upon the data. Given a vocabulary list some semantic measures combined with the phonetic measures can give better results. For identifying the cognates based on recurrent sound correspondences we need a larger word list for training the system. In this thesis we only handle the data where the word pairs are generated by taking all possible word pairs from two word lists.

Identifying cognates is not only important for historical linguistics but is also a very important task for statistical machine translation models [3], sentence alignment [73, 56, 21], induction of bilingual lexicon from bitexts [41] and word alignment [48]. In the context of bitext related tasks, 'cognate' usually refers to words with similar in form and meaning and donot make any distinction between borrowed and genetically related words. A recurring problem in cognate identification is 'false friends'. False friends are those which are phonetically and orthographically similar but donot have similar meaning and are not genetically related. We propose a new framework for solving this problem and use a distributional similarity based measure for identifying the (potential) cognates[2].

1.2 Letter to Phoneme Conversion

Letter-to-phoneme (L2P) conversion can be defined as the task of predicting the pronunciation of a word given its orthographic form [12].The pronunciation is usually represented as a sequence of phonemes. Letter-to-phoneme conversion systems play a very important role in spell checkers [82], speech synthesis systems [70] and transliteration [72]. Letter-to-phoneme conversion systems may also be effectively used for *cognate identification and transliteration.* The existing cognate identification systems use the orthographic form of a word as the input. But we know that the correspondence between written and spoken forms of words can be quite irregular as is the

[2] We donot address the problem of false friends in this thesis

case in English. Even in other languages with supposedly regular spellings, this irregularity exists owing to linguistic phenomena like borrowing and language variation. Letter-to-phoneme conversion systems can facilitate the task of cognate identification by providing a language independent transcription for any word.

Until a few years ago, letter-to-phoneme conversion was performed considering only one-one correspondences [15, 25]. Recent work uses many-to-many correspondences [38] and reports significantly higher accuracy for Dutch, German and French. The current state of the art systems give as much as 90% [37] accuracy for languages like Dutch, German and French. However, accuracy of this level is yet to be achieved for English.

A very important point that has to be observed is that whatever results one gets in this task are data dependent. In no way can one directly compare two systems which have been tested using different data sets. This poses a problem which we have not addressed in this paper, but it has to be kept in mind while comparing the results of our experiments with the previously reported results. Rule-based approaches to the problem of letter-to-phoneme conversion although appealing, are impractical as the number of rules for a particular language can be very high [43]. Alternative approaches to this problem are based on machine learning and make use of resources such as pronunciation dictionaries. In this paper, we present one such machine learning based approach wherein we envisage this problem as a Statistical Machine Translation (SMT) problem.

1.3 Phylogenetic Trees

Ever since the beginning of evolutionary thought, intuitions have been galore about the relevance of the process of evolution to Language change. However, in the field of linguistic theory itself, the idea of “a common origin” had existed long before Darwin’s observation of ‘curious parallels’ between the processes of biological and linguistic evolution. The birth of comparative philology as a methodology is often attributed to that now very well-known observation by Sir William Jones that there existed numerous similarities between far-removed languages such as Sanskrit, Greek, Celtic, Gothic and Latin which was impossible unless they had ‘sprung from some common source, which perhaps no longer exists’. This observation also marked the birth of the Indo-European language family hypothesis. Though Jones may not have been the first to suggest a link between Sanskrit and some of the European languages, it was only after his famous remarks that explanations for the enormous synchronic diversity of language started assuming a historical character. Up until that point in linguistic theory, explanations for the similarity and therefore, the relationship between different languages had been purely taxonomic and essentially ahistorical. See [8] for a very interesting account of the comparative study of the development of fields of linguistic and biological theory.

Language change came to be seen as a process of ‘descent with modification’ from a common origin. If one were to employ modern linguistic terminology following Saussure to characterize this earlier phase of linguistic research, it could be said that synchrony was sought to be understood via diachrony. And diachrony, it was for the next century or so of linguistic research during which time the philological method was at its peak until the arrival of Saussure. Although the Indo-European language family hypothesis came into existence soon after Jones’ findings, it was not until much later that the nature of the relationship between a set of languages was represented using a tree topology. The use of a tree topology to represent relationships among a set of languages made explicit the underlying idea of *linguistic diversification* which the Neogrammarian hypothesis entailed. ‘Linguistic diversification refers to how a single ancestor language (a proto-language) develops dialects which in time through the accumulation of changes become distinct languages.’ [19] In spite of its traditional importance in historical linguistics, the Family-tree model of language change has had quite a few criticisms directed at it, most notably for its neglect of the phenomenon of borrowing. The strongest challenge to the family-tree model which is based on the Neogrammarian hypothesis comes from dialectology in the form of a ’wave theory’ model of language change.

1.4 Contributions

- We have proposed a new framework called *Feature N-grams* for the identification of cognates which outperforms the orthographic measures such as edit distance, longest common subsequence ratio and dice. We use distributional similarity, the first attempt in this direction, to our knowledge, for identifying cognates.

- We prepared a list of cognates for Dravidian languages which can be used for further experiments. This is the first time any computational methods have been applied to Dravidian languages.

- We showed that the phrase based statistical machine translation system can be used for letter to phoneme conversion with results comparable to state-of-the-art systems.

- We showed that the data obtained in the framework for lexical diffusion can be used successfully as a input for phylogenetic methods from bioinformatics for construction of family trees. This work shows that if we can determine the words where the lexical diffusion of a single sound change is in place, it can be used effectively for constructing the language trees for a family or sub-family. Indeed the costly procedure of preparing bilingual word lists can be avoided in this process.

1.5 Outline

- Chapter 2 talks about the system designed and implemented for cognate identification. It begins with the description of the related work and then proceeds to describe the different baseline measures which we have used to evaluate our system. It describes the framework of feature n-grams and then describes the two methods which we designed and used for determing the similarity between a pair of words.

- Chapter 3 describes the related work on letter to phoneme conversion system, applications of the system, specifically to cognate identification and the mathematical foundations of this approach. We evaluate our system's efficiency for various languages and show that our system produces results which are comparable to the state-of-the-art.

- Chapter 4 begins with describing the related work in constructing phylogenetic trees. In this process we provide an overview of the basics and the similarity between the language evolution and biological evolution and the appropriateness in using measures from bioinformatics for the study of language change. The dataset and the languages under the focus of our study are briefly described in this chapter. We use a family of methods from bioinformatics and construct a tree which is very similar to the tree obtained by the traditional comparative method.

- Chapter 5 gives a conclusion and future work of the problems of cognate identification, letter to phoneme conversion and phylogenetic trees.

Chapter 2

Cognate Identification

2.1 Introduction

This chapter describes the first attempt in applying computational methods for identifying the cognates in Dravidian languages. We describe the related work in cognate identification and then describe the baseline measures. Next we describe our new framework and the similarity measures which we use for identifying the cognates. Finaly we discuss the results of our work.

2.2 Related Work

Kondrak [46] proposed algorithms for aligning two cognates, given the phonetic transcriptions, based on phonetic feature values. The system which he calls ALINE [44] assigns a similarity score to the two strings being compared. In another paper [45] he combines semantic similarity with the phonemic similarity to identify the cognates between two languages. Another major work of Kondrak is using the word alignment models from statistical machine translation for determining the sound correspondences between two word lists for related languages.

All the above works donot make any distinction between borrowings from true cognates. The algorithms also identify false friends between two related languages as cognates because of their phonetic or orthographic similarity. Identifying the borrowings is really a tough task as the borrowings seemingly look as a native word on the surface and much deeper linguistic knowledge is required to identify whether a word is a borrowing or not [1].

There has been some work done in identifying false friends from true cognates. Inkpen et al. [36] has used various machine learning algorithms for identifying false friends. Various orthographic similarity functions between English and French are used as features for training the machine learning algorithms. They achieve as high as 98% accuracy in identifying the false friends. Frunza et al. [33] use semi-supervised bootstrapping of semantic senses to identify the partial cognates between English and French. In another work Mulloni et al. [57] used sequence labeling techniques such as SVM (Support Vector Machines) for identifying cognates from written text without using any phonetic or semantic features. Bergsma et al. [13] use character-based alignment features as an input for the discriminative classifier for classifying the word pairs as cognates or non-cognates.

Its always interesting to know which methods perform well, orthographic methods or methods which use linguistic features (both phonetic and semantic). In this direction Kondrak et al. [49] evaluate various phonetic similarity algorithms for evaluating their effectiveness in identifying cognates. Their experiments show that orthographic measures indeed outperform manually constructed methods.

All the above work was done on Indo-European languages or Algonquian languages. In this thesis we make an effort to identify cognates for the Dravidian languages. The orthographic measures donot take the actual sounds represented by the alphabets into consideration but simply calculate the similarity of a word pair based on their character similarity. The phonetic measures take the features of the individual sounds into consideration for estimating the similarity between the words. The orthographic measures are usually used as a baseline against which any cognate identification system is tested. In this chapter we only take three such orthographic measures i.e. *Scaled Edit Distance, Dice, LCSR*. All these measures are explained in

[1] I have tried to use phonetic feature-value pairs as features for machine learning and tried to identify the origin of the words with some success. This is a problem which needs addressing separately and I believe can become the focus of an independent study by itself

the next section.

2.3 Orthographic Measures

Dice similarity was used previously for comparing biological sequences which is now being used for estimating word similarity. It is calculated by dividing twice the total number of shared letter bigrams by the sum of the total number of letter bigrams in both the words.

$$DICE(x, y) = \frac{2\,|bigrams(x) \cap bigrams(y)|}{|bigrams(x)| + |bigrams(y)|} \tag{2.1}$$

For example, DICE(*colour*,*couleur*) = 6/11 = 0.55 (the shared bigrams are *co*, *ou*, *ur*).

LCSR (Longest Common Subsequence Ratio) is computed by dividing the longest common subsequence by the length of the longer string. Melamed [56] has proposed that the if the similarity between two strings is greater than 0.58 than they can be cognates. For example, LCSR between *colour*,*couleur* is = 5/7 = 0.71.

Scaled Edit Distance (SED) is the scaled edit distance. The edit distance is calculated by the minimum edits required to transform one string to another. The edit operations are substitutions, insertions and deletions all with a cost of 1. The edit distance is normalised by the average of the lengths of the two strings under comparision.

2.4 Feature N-grams

The idea in using this measure is that the way phonemes occur together matters less than the way the phonetic features occur together because phonemes themselves are defined in terms of the features. Therefore, it makes more sense to a have measure directly in terms of phonetic features. But since we are experimenting directly with corpus data (without any phonetic transcription) using the CPMS [75], we also include some orthographic features as given in the CPMS implementation. The letter to feature mapping that we use comes from the CPMS. Basically, each word is converted into a set of sequences of feature-value pairs such that any feature can follow any feature, which means that the number of sequences for a word of length l_w is less than or equal to $(N_f \times N_v)^{l_w}$, where N_f is the number of possible features and N_v is the number of possible values. We create sequences of feature-value pairs for each word and from this 'corpus' of feature-value pair sequences we build the feature n-gram model.

The feature n-grams are computed as follows. For a given word, each letter is first converted into a vector consisting of the feature-value pairs which are mapped to it by the CPMS. Then, from the sequence of vectors of features, all possible sequences of features up to the length 3 (the order of the n-gram model) are computed. All these sequences of features (feature n-grams) are added to the n-gram model. Finally the model is pruned as mentioned above. We expected this measure to work better because it works at a higher level of abstraction and is more linguistically valid. *Method 1* is based on distributional similarity, whereas *Method 2* is based on the feature n-gram version of DICE. Details about the two methods are in the next paragraph.

Method 1

For a given word pair, feature-value n-grams and their corresponding probabilities are estimated for each word by treating each word as small corpus and compiling feature-value based n-gram model. For each word, all the n-grams irrespective of their sizes (unigram, bigram etc.) are merged in one vector, as mentioned earlier. Now that we have two probability distributions, we can calculate how similar they are using any information theoretic or distributional similarity measure. For our experiments, we used normalized symmetric cross entropy as given in eqn. 2.2.

$$d_{sce} = \sum_{g_l = g_m} (p(g_l)\ log\ q(g_m) + q(g_m)\ log\ p(g_l)) \qquad (2.2)$$

The formula for calculating distributional similarity based on these phonetic and orthographic features is the same (SCE) as given in equation 2.2, except that the distribution in this case is made up of features rather than letters. Note that since we do not assume the features to be independent, any feature can follow any other feature in a feature n-gram. All the permutations are computed before the feature n-gram model is pruned to keep only the top N feature n-grams. The order of the n-gram model is kept as 3, i.e., trigrams.

2.5 Experimental Setup

The data for this experiment was obtained from Dravidian Etymological Dictionary[2]. Word lists for Tamil and Malayalam were extracted from the dictionary. Only the first 500 entries in each word list were manually verified. The candidate pair set was created by generating all the possible Tamil-Malayalam word pairs. The electronic version of the dictionary was used as the gold standard. The task was to identify 329 cognate pairs out of the 250,000 candidate pairs (0.1316%). The standard string similarity measures such as Scaled Edit Distance (SED), Longest Common Subsequence Ratio (LCSR) and the Dice measures were used as baselines for the experiment. The system was evaluated using *11-point interpolated average precision* [54]. The candidate pairs are reranked based on the similarity scores calculated for each candidate pair. The 11-point interpolated average precision is an information extraction evaluation technique. The precision levels are calculated for the recall levels of 0%, 10%, 20%, 30%,.....,100%, and then averaged to a single number. The precision at recall levels 0% and 100% are uniformly set at 1 and 0 respectively.

	SED	LCSR	DICE	*Feature-Value n-Gram*	FNGDICE
Genetic Cognates	49.32%	52.02%	51.06%	53.98%	60%

Table 2.1: Results for cognate identification using distributional similarity for feature-value pair based model as compared to some other sequence similarity based methods

[2] http://dsal.uchicago.edu/cgi-bin/philologic/getobject.pl?c.0:1:3.burrow

2.6 Results

The results for the four measures are given in the Table 2. The precision is highest for feature-value pair based n-grams, inspite of the fact that the measure used by us is a distributional similarity measure, whereas the other three are sequence similarity measure. We have not yet performed experiments using sequence probability given the model of phonetic space, but intuitively the result for sequence probability should be better than for distributional similarity because we are trying to compare two sequences, not two distributions. Still, the results do show that feature-value based model can outperform phoneme based model for certain applications.

Chapter 3

Modeling Letter-to-Phoneme Conversion as a Phrase Based Statistical Machine Translation Problem with Minimum Error Rate Training

3.1 Introduction

The outline of this chapter is as follows. Section 3.2 presents a brief summary of the related work done in L2P conversion. Section 3.3 describes our model and the techniques devised for optimizing the performance. Section 3.4 describes the letter-to-phoneme alignment. The description of the results and experiments and a new technique for estimating the difficulty level of L2P task have been given in Section 3.5. Error analysis is presented in Section 3.6. Finally we conclude with a summary and suggest directions for future work.

3.2 Related Work

In the letter-to-phoneme conversion task, a single letter can map to multiple phonemes [x → ks] and multiple letters can generate a single phoneme. A letter can also map to a null phoneme [e → φ] and vice-versa. These examples give a glimpse of why the task is so complex and a single machine learning technique may not be enough to solve the problem. A overview of the literature supports this claim.

In older approaches, the alignment between the letters and phonemes was taken to be one-to-one [15] and the phoneme was predicted for every single letter. But recent work [14, 38] shows that multiple letter-to-phoneme alignments perform better than single letter to phoneme alignments. The problem can be either viewed as a multi-class classifier problem or a structure prediction problem. In structure prediction, the algorithm takes the previous decisions as the features which influence the current decision.

In the classifier approach, only the letter and its context are taken as features. Then, either multiclass decision trees [24] or instance based learning as in [84] is used to predict the class, which in this case is a phoneme. Some of these methods [15] are not completely automatic and need an initial handcrafted seeding to begin the classification.

Structure prediction is like a tagging problem where HMMs [81] are used to model the problem. Taylor claims that except for a preprocessing step, it is completely automatic. The whole process is performed in a single step. The results are poor, as reasoned in [37] due to the emission probabilities not being informed by the previous letter's emission probabilities. Pronunciation by Analogy (PbA) is a data-driven method [55] for letter-to-phoneme conversion which is used again by Damper et al [25]. They simply use an Expectation-Maximisation (EM) like algorithm for aligning the letter-phoneme pairs in a speech dictionary. They claim that by integrating the alignments induced by the algorithm into the PbA system, they were able to improve the accuracy of the pronunciation significantly. We also use the many-to-many alignment approach but in a different way and obtained from a different source.

The recent work of Jiampojamarn et al [38] combines both of the above approaches in a very interesting manner. It uses an EM like algorithm for aligning the letters and phonemes. The algorithm allows many-to-many alignments between letters and phonemes. Then there is a letter chunking module which uses instance-based training to train on the alignments which have been obtained in the previous step. This module is used to guess the possible letter chunks in every word. Then a local phoneme predictor is used to guess the phonemes for every letter in a word. The size of the letter chunk could be either one or two. Only one candidate for every word is allowed. The best phoneme sequence is obtained by using Viterbi search.

An online model MIRA [23] which updates parameters is used for the L2P task by Jiampojamarn et al [37]. The authors unify the steps of letter segmentation, phoneme prediction and sequence modeling into a single module. The phoneme prediction and sequence modeling are considered as tagging problems and a Perceptron HMM [22] is used to model it. The letter segmenter module is replaced by a monotone phrasal decoder [85] to search for the possible substrings in a word and output the n-best list for updating MIRA. Bisani and Ney [14] take the joint multigrams of graphemes and phonemes as features for alignment and language modeling for phonetic transcription probabilities. A hybrid approach similar to this is by [83].

In the next section we model the problem as a Statistical Machine Translation (SMT) task.

3.3 Modeling the Problem

Assume that given a word, represented as a sequence of letters $\mathbf{l} = l_1^J = l_1...l_j...l_J$, needs to be transcribed as a sequence of phonemes, represented as $\mathbf{f} = f_1^I = f_1...f_i...f_I$. The problem of finding the best phoneme sequence among the candidate translations can be represented as:

$$\mathbf{f}_{best} = \arg\max_{\mathbf{f}} \{\Pr(\mathbf{f} \mid \mathbf{l})\} \tag{3.1}$$

We model the problem of letter to phoneme conversion based on the noisy channel model. Reformulating the above equation using Bayes Rule:

$$\mathbf{f}_{best} = \arg\max_{\mathbf{f}} p\left(\mathbf{l} \mid \mathbf{f}\right) p\left(\mathbf{f}\right) \tag{3.2}$$

This formulation allows for a phoneme n-gram model $p\left(\mathbf{f}\right)$ and a transcription model $p\left(\mathbf{l} \mid \mathbf{f}\right)$. Given a sequence of letters $\mathbf{l}$, the argmax function is a search function to output the best phonemic sequence. During the decoding phase, the letter sequence $\mathbf{l}$ is segmented into a sequence of K letter segments $\bar{l}_1^K$. Each segment $\bar{l}_k$ in $\bar{l}_1^K$ is transcribed into a phoneme segment $\bar{f}_k$. Thus the best phoneme sequence is generated from left to right in the form of partial translations. By using an n-gram model p_{LM} as the language model, we have the equations:

$$\mathbf{f}_{best} = \arg\max_{\mathbf{f}} p\left(\mathbf{l} \mid \mathbf{f}\right) p_{LM} \tag{3.3}$$

with $p\left(\mathbf{l} \mid \mathbf{f}\right)$ written as

$$p(\bar{l}_1^K \mid \bar{f}_1^K) = \prod_{k=1}^{K} \Phi(\bar{l}_k \mid \bar{f}_k) \tag{3.4}$$

From the above equation, the best phoneme sequence is obtained based on the product of the probabilities of transcription model and the probabilities of a language model and their respective weights. The method for obtaining the transcription probabilities is described briefly in the next section. Determining the best weights is necessary for obtaining the right phoneme sequence. The estimation of the models' weights can be done in the following manner.

The posterior probability $\Pr\left(\mathbf{f} \mid \mathbf{l}\right)$ can also be directly modeled using a log-linear model. In this model, we have a set of M feature functions $h_m(\mathbf{f},\, \mathbf{l}), m = 1...M$. For each feature function there exists a weight or model parameter $\lambda_m, m = 1...M$. Thus the posterior probability becomes:

$$\Pr\left(\mathbf{f} \mid \mathbf{l}\right) = p_{\lambda_1^M}\left(\mathbf{f} \mid \mathbf{l}\right) \tag{3.5}$$

$$= \frac{\exp\left[\Sigma_{m=1}^{M} \lambda_m h_m(\mathbf{f},\ \mathbf{l})\right]}{\sum_{f_1^I} \exp\left[\Sigma_{m=1}^{M} \lambda_m h_m(\acute{f}_1^I, \mathbf{l})\right]} \tag{3.6}$$

with the denominator, a normalization factor that can be ignored in the maximization process.

The above modeling entails finding the suitable model parameters or weights which reflect the properties of our task. We adopt the criterion followed in [60] for optimising the parameters of the model. The details of the solution and proof for the convergence are given in Och [60]. The models' weights, used for the L2P task, are obtained from this training.

3.4 Letter-to-Phoneme Alignment

We used GIZA++ [61], an open source toolkit, for aligning the letters with the phonemes in the training data sets. In the context of SMT, say English-Spanish, the parallel corpus is aligned bidirectionally to obtain the two alignments. The IBM models give only one-to-one alignments between words in a sentence pair. So, GIZA++ uses some heuristics to refine the alignments [61].

In our input data, the source side consists of grapheme (or letter) sequences and the target side consists of phoneme sequences. Every letter or grapheme is treated as a single 'word' for the GIZA++ input. The transcription probabilities can then be easily learnt from the alignments induced by GIZA++, using a scoring function [42]. Figure 3.1 shows the alignments induced by GIZA++ for the example words which are mentioned by Jiampojamarn et al [38]. In this figure, we only show the alignments from graphemes to phonemes.

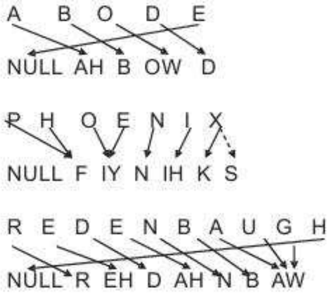


Figure 3.1: Example Alignments from GIZA++

3.5 Evaluation

We evaluated our models on the English CMUDict, French Brulex, German Celex and Dutch Celex speech dictionaries. These dictionaries are available for download on the website of PROANALSYL[1] Letter-to-Phoneme Conversion Challenge. Table 3.1 shows the number of words for each language. The datasets available at the website were divided into 10 folds. In the process of preparing the datasets we took one set for test, another for developing our parameters and the remaining 8 sets for training. We report our results in word accuracy rate, based on 10-fold cross validation, with mean and standard deviation.

Language	Datasets	Number of Words
English	CMUDict	112241
French	Brulex	27473
German	Celex	49421
Dutch	Celex	116252

Table 3.1: Number of words in each Dataset

We removed the one-to-one alignments from the corpora and induced our own alignments using GIZA++. We used minimum error rate training [60] and the A* beam search decoder implemented by Koehn [42]. All the above tools are available as parts of the MOSES [40] toolkit.

3.5.1 Exploring the Parameters

The parameters which have a major influence on the performance of a phrase-based SMT model are the alignment heuristics, the maximum phrase length (MPR) and the order of the language model [42]. In the context of letter to phoneme conversion, *phrase* means a sequence of letters or phonemes mapped to each other with some probability (i.e., the *hypothesis*) and stored in a phrase table. The *maximum phrase length* corresponds to the maximum number of letters or phonemes that a hypothesis can contain. Higher phrase length corresponds a larger phrase table during decoding.

[1] http://www.pascal-network.org/Challenges/PRONALSYL/

We have conducted experiments to see which combination gives the best output. We initially trained the model with various parameters on the training data and tested for various values of the above parameters. We varied the maximum phrase length from 2 to 7. The language model was trained using SRILM toolkit [77]. We varied the order of language model from 2 to 8. We also traversed the alignment heuristics spectrum, from the parsimonious *intersect* at one end of the spectrum through *grow, grow-diag, grow-diag-final, grow-diag-final-and* and *srctotgt* to the most lenient *union* at the other end. Our intuitive guess was that the best alignment heuristic would be *union.*

We observed that the best results were obtained when the language model was trained on 6-gram and the alignment heuristic was *union.* No significant improvement was observed in the results when the value of MPR was greater than 5. We have taken care such that the alignments are always monotonic. Note that the average length of the phoneme sequence was also 6. We adopted the above parameter settings for performing training on the input data.

3.5.2 System Comparison

We adopt the results given in [38] as our baseline. We also compare our results with some other recent techniques mentioned in the Related Work section. Table 3.2 shows the results. As this table shows, our approach yields the best results in the case of German and Dutch. The word accuracy obtained for the German Celex and Dutch Celex dataset using our approach is higher than that of all the previous approaches listed in the table. In the case of English and French, although the baseline is achieved through our approach, the word accuracy falls short of being the best. However, it must also be noted that the dataset that we used for English is slightly larger than those of the other systems shown in the table.

We also observe that for an average phoneme accuracy of 91.4%, the average word accuracy is 63.81%, which corroborates the claim by Black et al [15] that a 90% phoneme accuracy corresponds to 60% word accuracy.

Language	Dataset	Baseline	1-1 Align	1-1 + CSIF	1-1 + HMM	M-M Align	M-M + HMM	MeR + A*
English	CMUDict	58.3±0.49	60.3±0.53	62.9±0.45	62.1±0.53	65.1±0.60	65.6±0.72	63.81±0.47
German	Celex	86.0±0.40	86.6±0.54	87.6±0.47	87.6±0.59	89.3±0.53	89.8±0.59	90.20±0.25
French	Brulex	86.3±0.67	87.0±0.38	86.5±0.68	88.2±0.39	90.6±0.57	90.9±0.45	86.71±0.52
Dutch	Celex	84.3± 0.34	86.6±0.36	87.5±0.32	87.6±0.34	91.1±0.27	91.4±0.24	91.63±0.24

Table 3.2: System Comparison in terms of word accuracies. **Baseline:**Results from PRONALSYS website. **CART:** CART Decision Tree System [15]. **1-1 Align, M-M align, HMM:** one-one alignments, many-many alignments, HMM with local prediction [38]. **CSIF:**Constraint Satisfaction Inference(CSIF) of[83]. **MeR+A*:**Our approach with minimum error rate training and A* search decoder. "-" refers to no reported results.

3.5.3 Difficulty Level and Accuracy

We also propose a new language-independent measure that we call 'Weighted Symmetric Cross Entropy' (WSCE) to estimate the difficulty level of the L2P task for a particular language. The *weighted* SCE is defined as follows:

$$d_{sce_{wt}} = \sum r_t \; (p_l \; log \; (q_f) + q_f \; log \; (p_l)) \tag{3.7}$$

where p and q are the probabilities of occurrence of letter (l) and phoneme (f) sequences, respectively. Also, r_t corresponds to the conditional probability $p(f \mid l)$. This transcription probability can be obtained from the phrase tables generated during training. The weighted entropy measure $d_{sce_{wt}}$,for each language, was normalised with the total number of such n-gram pairs being considered for comparison with other languages. We have fixed the maximum order of l and f n-grams to be 6. Table 3.3 shows the difficulty levels as calculated using WSCE along with the accuracy for the languages that we tested on. As is evident from this table, there is a rough correlation between the difficulty level and the accuracy obtained, which also seems intuitively valid, given the nature of these languages and their orthographies.

Language	Datasets	$d_{sce_{wt}}$	Accuracy
English	CMUDict	0.30	63.81±0.47
French	Brulex	0.41	86.71±0.52
Dutch	Celex	0.45	91.63±0.24
German	Celex	0.49	90.20±0.25

Table 3.3: $d_{sce_{wt}}$ values predict the accuracy rates.

3.6 Error Analysis

In this section we present a summary of the error analysis for the output generated. We tried to observe if there exist any patterns in the words that were transcribed incorrectly. The majority of errors occurred in the case of vowel transcription, and diphthong transcription in particular. In the case of English, this can be attributed to the phenomenon of lexical borrowing from a variety of sources as a result of which the number of sparse alignments is very high. The system is also unable to learn allophonic variation of certain kinds of consonantal phonemes, most notably fricatives like /s/ and /z/. This problem is exacerbated by the irregularity of allophonic variation in the language itself.

Chapter 4

An Application of Character Methods for Dravidian Languages

4.1 Introduction

The outline of the chapter is as follows. Section 4.2 gives the basics and background of the various terms used in bioinformatics for infering phylogenetic trees and their parallels in historical linguistics. Section 4.3 describes the dataset used in our experiments.Section 4.4 and 4.5 describes the distance methods and the results of the experiments. Section 4.6 describes the character based methods and the results. Finally the chapter concludes with the discussion of the trees resulting from the experiments.

4.2 Basics and Related Work

Once glottochronology[1] was hugely popular for constructing family tree and estimating divergence times which are no longer popular. In recent years, the methods developed in computational biology were used for inferring phylogenetic trees. Based on the similarity between language evolution and biological evolution the methods have been successfully applied to languages for constructing the phylogeny. All these methods are character based or distance based methods. The availability of data sets for well-established language families like Indo-European [27] has spurred a number of researchers to apply these methods to these data sets and validate the resultant phylogenetic trees against the well-established linguistic facts and to test competing hypotheses. We give a overview of the terminology used in the following section.

[1] A major attempt to construct family trees and estimate the language divergence times was previously done using lexicostatistics and glottochronology. Lexicostatistics was introduced by Morris Swadesh [79]. A list of cognate words in the languages being analysed is used to build a family tree. In the first step a basic meaning list is taken which is supposed to be resistant to borrowing and replacement and the meanings are supposed to be culturally-free and universal. Concepts such as body parts, numerals, elements of nature etc. are present in the list. The idea is that no human language would be complete without this list. Once such a meaning list is composed, the common words in each language is used to fill the list. In the second step the cognates among these words are found by using comparative method. Any borrowings are discarded from the list. In the third step the distance between each pair of languages is supposed to be the number of shared cognates between the corresponding pair. By using a technique called UPGMA[2] the distances are used to construct a family tree for the languages.

Now glottochronology is used to estimate the divergence time for each node in the family tree. Glottochronology has the assumption that the rate of lexical replacement is constant for all languages at all times. This constant is called as glottochronological constant and the value is fixed at 0.806. Swadesh [79] used the following formula for estimating the divergence times of Amerindian languages where r is the glottochronological constant and c is the percentage of shared cognates.

$$t = \frac{log\ c}{2\ log\ r} \tag{4.1}$$

The glottochronology method has been criticised for the following reasons. First, there is a loss of information when the character-state data is converted to percentage similarity scores. Second, the problem that a language can have multiple words, may or may not have a word is not handled. Third, the rate of evolution among languages is quite different and the assumption of a universal rate constant doesnot hold. Fourth, the UPGMA method based on the percentage of shared cognates can produce inaccurate branch lengths and thus produce erroneous divergence times. Also the language evolution is not always tree-like. For this reasons the researchers in the last 10 years started using techniques from bioinformatics to infer phylogenetic trees.

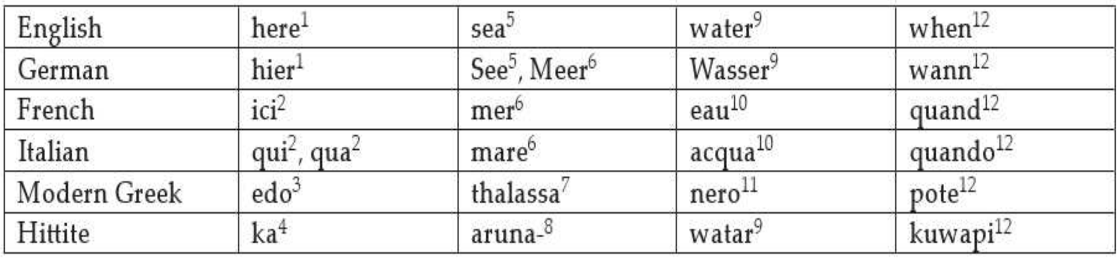
4.2.1 Basic Concepts

Characters

Language evolution can be seen as a change in some of its features. A character encodes the similarity between the languages on the basis of these features and defines a equivalence relation on the set of languages L. Defining the character formally

> A character is a function $c : L \rightarrow Z$ where L is the set of languages and Z is the set of integers.

A character can take different forms across a set of languages which are called “states”. These characters can either be lexical, phonological or morphological features. The actual values of these characters are not important [65]. A **lexical** character corresponds to a meaning slot. For a given meaning, lexical items for different languages fall into different cognate classes (based on the cognacy judgment between them) and different cognate classes form the different states of the character. Two languages would have same state if they have lexical items which are cognates. Figure 4.1 shows an example of how the lexical characters are represented for a meaning slot. The superscript shows the state exhibited by each language for a particular meaning slot. **Morphological** characters are normally inflectional markers and are coded by cognation like lexical items. **Phonological** characters are used to represent the presence or absence of particular sound change(or a series of sound changes) in the corresponding language.



English	here^{1}	sea^{5}	water^{9}	when^{12}
German	hier^{1}	See^{5}, Meer^{6}	Wasser^{9}	wann^{12}
French	ici^{2}	mer^{6}	eau^{10}	quand^{12}
Italian	qui^{2}, qua^{2}	mare^{6}	acqua^{10}	quando^{12}
Modern Greek	edo^{3}	thalassa^{7}	nero^{11}	pote^{12}
Hittite	ka^{4}	aruna-^{8}	watar^{9}	kuwapi^{12}

Figure 4.1: Consensus tree of Indo-European languages obtained by Gray and Atkinson (2003) using penalized maximum likelihood on lexical items.

Homoplasy and Perfect Phylogenies

Two languages can share the same state not only due to shared evolution but also due to phenomena called **backmutation** and **parallel development**. These phenomena are jointly referred to as **homoplasy**. For a particular character, if the already observed state reappears in the tree then the phenomenon is called backmutaion. Two languages can independently evolve in a similar fashion. In that case the two languages exhibit the same state which is called as parallel development. All of the initial work has assumed homoplasy-free evolution. When a character evolves without homoplasy down the tree then it is said to be compatible for that tree and the tree is said to be a **perfect phylogeny**. Hence everytime the character's state changes all the subtrees rooted at that point share the same state. Another source of ambiguity in the states of a character can be due to borrowing and are normally discarded.

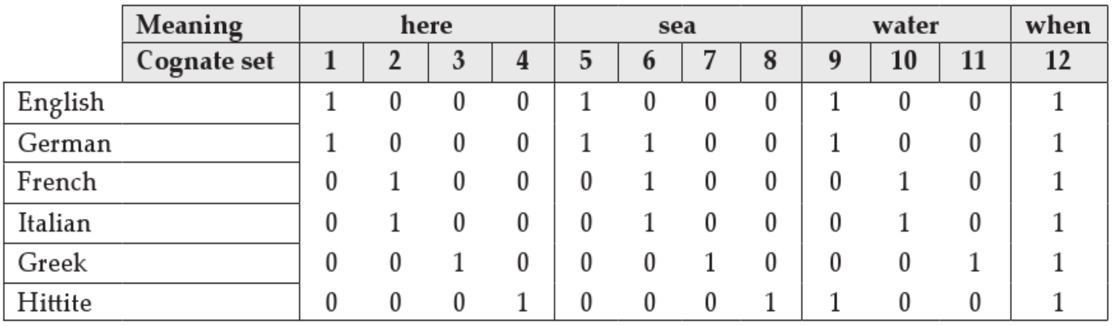
4.2.2 Related Work

The fashion in which characters evolve down the tree is described by a model of evolution. This specification or non-specification of models of evolution broadly divide the phylogenetic inference methods into two categories. For example the methods such as Maximum Parsimony, Maximum Compatibility and Distance methods such as Neighbour Joining and UPGMA donot require a explicit model of evolution. But statistical methods like Maximum Likehood and Bayesian Inference are parametric methods where the parameters of the model are tree topology, branch length and the rates of variation across sites. There is an interesting debate is going on in the scientific community regarding the appropriateness of the assumption of a model of evolution for linguistic data [30].

Gray and Jordan were among the first to apply Maximum Parsimony to Austronesian language data. They applied the technique to 5,185 lexical items from 77 Austronesian languages and were able to get a single most parsimonious tree. The maximum parsimony method returns the tree on which the minimum number of character state changes have taken place. There are different types of parsimonies such as Wagner, Camin-Soakal which have different assumptions about the character state changes. The assumptions of the above parsimonies is described in detail in the section 4.6.

Particularly interesting is the work of Gray and Atkinson [7, 9] who applied bayesian inference techniques [35] to the Indo-European database. They used a binary valued matrix to represent the lexical characters. Although their tree had nothing new in terms of its structure, it was identical to the tree established by the historical linguists (the position of Albanian not resolved), the dating based on *penalised likelihood* supported the famous Anatolian hypothesis compared to Krugan hypothesis, dating the Indo-European family as being 8000 years old. Their model assumes that the cognate sets evolve independently, they use a gamma distribution to model the variation across the cognate sets and try to find a sample of trees which matches their data. Unlike the other non-parametric methods mentioned above their method can handle polymorphism. By representing the cognate information in terms of binary matrices ,unlike glottochronology, the information is retained in this model. The idea was to test the model in the scenarios where the cognacy judgements were not completely accurate and where the model misspecification could cause a bias in the estimate. The model was tested on a different set of ancient data prepared by Ringe et al [65]. They further tested their model on synthetic data giving chance for borrowing to occur between different lineages. The model was tested against two kinds of borrowing viz- borrowing between any two lineages and borrowing between lineages which are located locally. The dating in all the above cases was largely consistent with the dating they had obtained on the Dyen's dataset, which they claim, upholds the robustness of the model.

Ryder [67] in his work used syntactic features as characters and applied the above methods for constructing the phylogenetic tree for Indo-European languages. He also used the same techniques for various language family data for grouping related languages into their respective language families. The syntactic features were obtained from WALS database [10]. The assumption was that the rate by which syntactic features are replaced through borrowing is much lesser than in the case of lexical items.



Meaning	here				sea				water			when
Cognate set	1	2	3	4	5	6	7	8	9	10	11	12
English	1	0	0	0	1	0	0	0	1	0	0	1
German	1	0	0	0	1	1	0	0	1	0	0	1
French	0	1	0	0	0	1	0	0	0	1	0	1
Italian	0	1	0	0	0	1	0	0	0	1	0	1
Greek	0	0	1	0	0	0	1	0	0	0	1	1
Hittite	0	0	0	1	0	0	0	1	1	0	0	1

Figure 4.2: An example of the binary matrix used by Gray and Atkinson.

Ringe et al [65] proposed a computational technique called **Maximum Compatibility** for constructing phylogenetic trees. The technique seeks to find the tree on which the highest number of characters are compatible. Their model assumes that the lexical data is free of **back mutation and parallel development**. The method was applied to a set of 24 ancient and modern Indo-European language data. They use morphological, lexical and phonological characters for inferring the phylogeny of these languages. Nakhleh et al [58] propose an extension to the method of Ringe et al known as Perfect Phylogenetic Networks which models homoplasy and borrowing explicitly. For a comparision of various phylogenetic methods on the ancient Indo-European data, refer [59]. They observed that almost all the methods except UPGMA had great similarity as well as striking differences between the trees. It must be noted that these scholars have not sought answers to much-disputed questions in the literature on the Indo-European language family tree such as the status of Albanian in their afore-mentioned quantitative analyses. In each of the attempts discussed till now, the main thrust has been to demostrate that language phylogeny as inferred using these quantitative methods was in almost perfect agreement with the traditional comparative method-based family tree thus demonstrating the utility of quantitative methods in the study of language change.

Ellison et al [28] discuss establishing a probability distribution for every language through intra-lexical comparison using confusion probabilities. They use scaled edit distance[3] to calculate the probabilities. Then the distance between every language is

[3] The edit distance between **by** and **rest** is 6.0 and between **interested** and **rest** is 6.0. Although

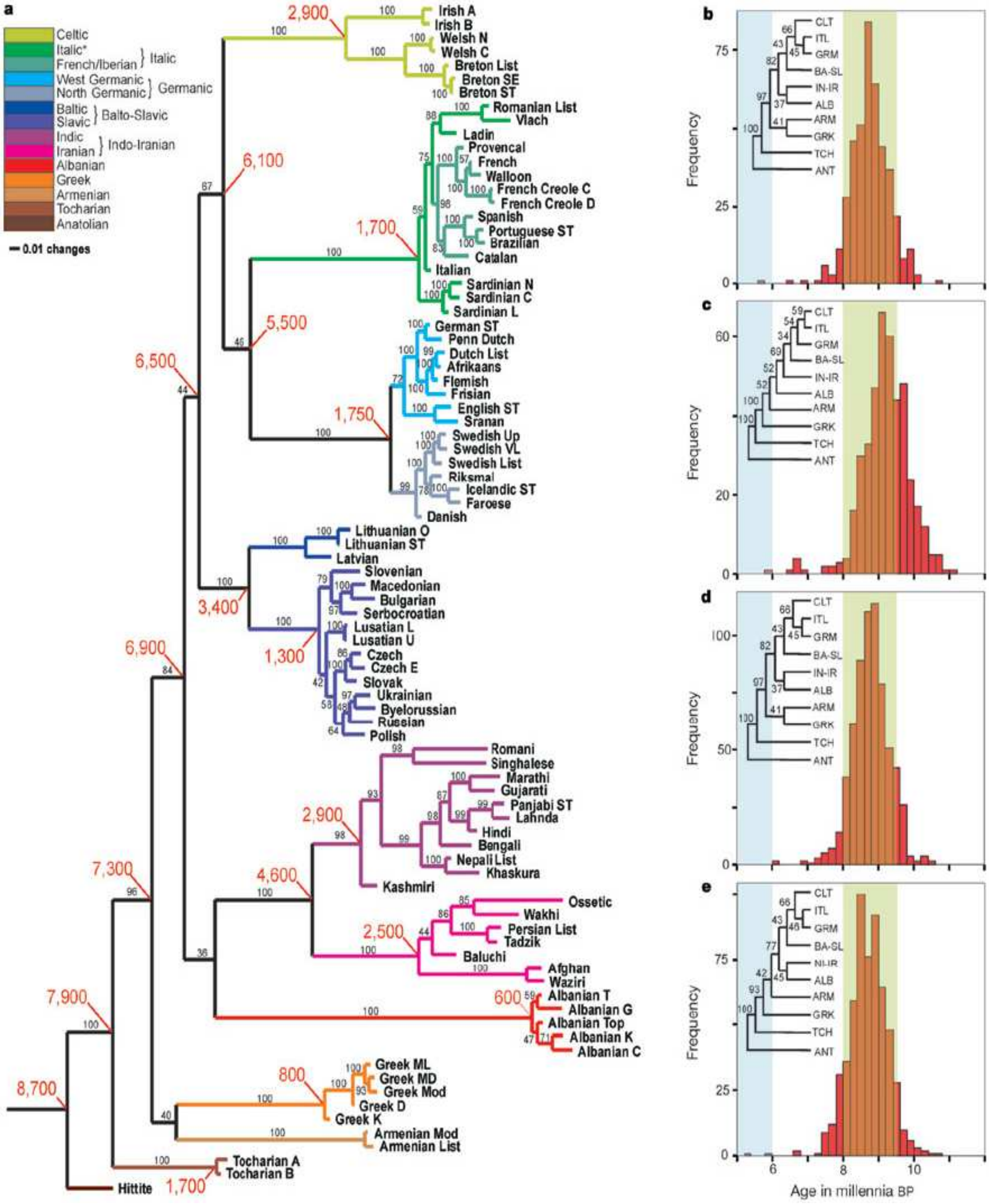


Figure 4.3: Consensus tree of Indo-European languages obtained by Gray and Atkinson (2003) using penalized maximum likelihood on lexical items.

estimated through KL-divergence and Rao's distance. The same measures are also used to find the level of cognacy between the words. The experiments are conducted on Dyen's [27] classical Indo-European dataset. The estimated distances are used for constructing the phylogeny of the Indo-European languages. Figure 4.4 shows the tree obtained using their method.

Alexandre Bouchard et al [17, 18] in a novel attempt, combine the advantages of the classical comparative method and the corpus-based probablistic models. The word forms are represented by phoneme sequences which undergo stochastic edits along the branches of a phylogenetic tree. The robustness of this model is tested against different tree topologies and it selects the linguistically attested phylogeny. Their stochastic model successfully models the language change by using synchronic languages to reconstruct the word forms in Vulgar Latin and Classical Latin. Although it reconstructs the ancient word forms of the Romance Languages, a major disadvantage of this model is that some amount of data of the ancient word forms is required to train the model, which may not be available in many cases.

Some earlier attempts by Andronov [5] using glottochronology for dating the Dravidian language family divergences was criticised for the largely faulty data used by him which made the dating unreliable and untenable. Krishnamurti et al [52] used unchanged cognates as a criterion for the subgrouping of South-Central Dravidian languages. Krishnamurti [50] prepared a list of 63 cognates in all the six languages which he determined would be sufficient for inferring the language tree of the family. They examined a total of 945 rooted binary trees[4] and apply the 63 cognates to every tree and then rank the trees. The tree which had the least score was considered to be the one that best represented the family tree.

both pairs have the same distance the first pair has nothing in common. The scaled edit distance is obtained by divding the distance by the average of the lengths of the two words. This makes the distance between the first pair to be 2.0 and the second pair to be 0.86.

[4] $(2n-3)/2^{n-2}(n-2)!$

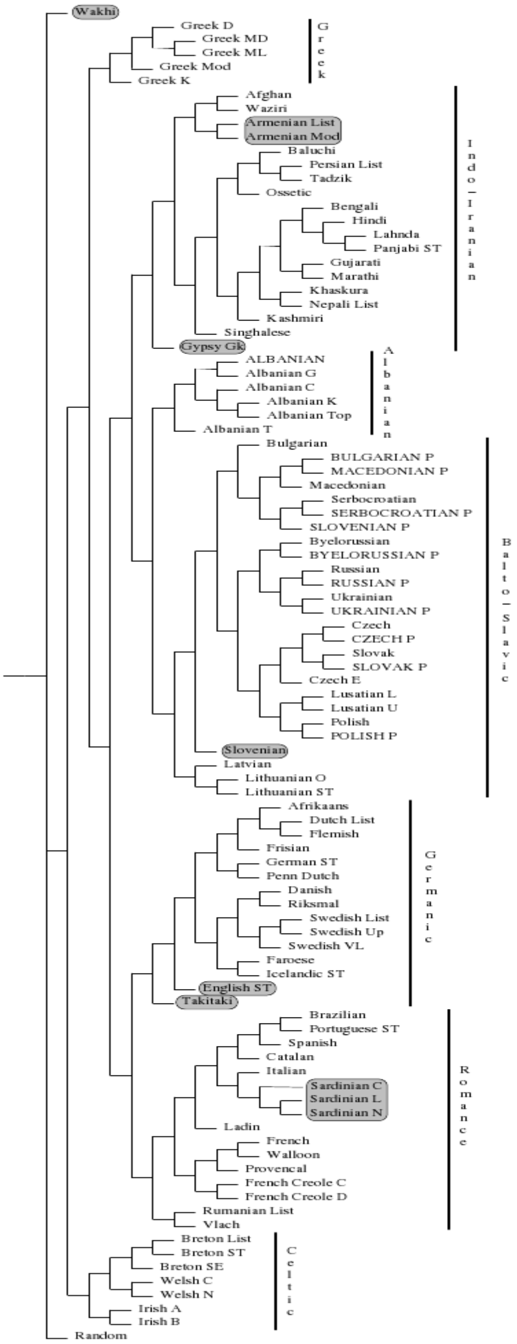


Figure 4.4: Tree of Indo-European Languages obtained using Intra-Lexical Comparision of Ellison and Kirby(2007)

4.3 Dataset

We used two different set of data for our experiments. The data is taken for the six South-Central (Now referred to as South Dravidian II in the recent literature. Refer to [51].) group of Dravidian Languages - viz. Gondi, Konda, Kui, Kuvi, Pengo, Manda. The data for the distance methods was obtained using the number of changed cognates every language pair shares. The number of shared cognates-with-change is the measure of the relative distance between the language pair. The following table shows the number of shared cognates between these languages (Taken from [52]).

The second data set was taken from Krishnamurti 1983 who provided the list of such cognates which were affected or not affected by sound change. We represented the unchanged cognates with 0 and changed cognates with 1. We use the same notation throughout the paper. We provide the dataset so that anyone can use the dataset and can replicate these experiments. This dataset was used as the input for character based methods.

Upto this point the literature which we have refered and mentioned in the section 4.2 use just the presence or absence of the sound change for infering phylogenetic trees and relationship between languages. Only those sound changes are taken which are supposed to be free of homoplasy. In this paper, we take the presence or absence of unchanged cognates as characters for inferring phylogenetic trees which we believe is a novel approach and has not been attempted before.

4.4 Distance Methods

All the distance based methods take the distance between two taxa as input and try to give the tree which explains the data. The assumption of a lexical clock may or may not hold depending upon the method. In our study we examine two such methods which are very popular in evolutionary biology and are also widely used in historical linguistics.

UPGMA (Unweighted Pair Group Method with Arithmetic Mean)

The lexicostatistics experiment for IE languages by [27] uses this method for the construction of the phylogenetic trees. The method works as follows.

1. Find the two closest languages (L1, L2) based on percentage of shared cognates.
2. Make L1,L2 siblings.
3. Remove one of them, say L1 from the set.
4. Recursively construct the tree on the remaining languages.
5. Make L1 the sibling of L2 in the final tree.

UPGMA assumes a uniform rate of evolution throughout the tree i.e, the distance of the root node to the leaves is equal. Moreover it produces a rooted tree whose ancestor is known.

Neighbour Joining (NJ)

Neighbour Joining is a type of agglomerative clustering method developed by Saitou and Nei [69]. It is also a greedy method like UPGMA but doesnot assume a uniform lexical clock hypothesis. Moreover the method produces unrooted trees with branch lengths which need to be rooted for inferring the ancestral states and the divergence times between the languages. The method starts out with a star-like topology and then tries to minimize an estimate of the total length of the tree by combining together the languages that provide the most reduction. It has been shown that the method is statistically consistent (if there is a tree which fits the lexical data perfectly, it retrieves the tree). The general observation is that Neighbour Joining returns the best tree out of all the distance based methods. There are other distance based methods such as FITSCH which are relatives (a generalised version) of UPGMA and NJ which we don't take up in our current study.

4.5 Experiments and Results for distance methods

Using a technique called U-statistic hierarchial clustering Roy D'Andrade [26] has used the above data and gave the following tree structure. The following tree structure in figure 4.5 exactly matches the tree given by Krishnamurti using morphological and

	Gondi	Konda	Kui	Kuvi	Pengo
Konda	16				
Kui	18	18			
Kuvi	22	20	88		
Pengo	11	19	48	49	
Manda	10	9	40	42	57

Table 4.1: Matrix of shared cognates-with-change

phonological isoglosses. For our purpose the similarity matrix in Table 4.1 is converted into a distance matrix using the following formula $d = 1/s_{ij}, i <= j$.

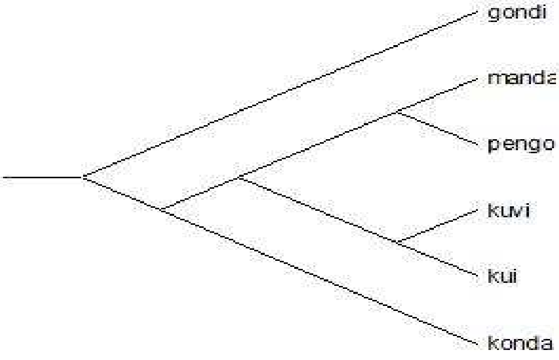


Figure 4.5: Tree obtained through comparative method

Figures 4.6 and 4.7 show the trees obtained by applying UPGMA and NJ methods on the data given in table 4.1.

4.6 Character Methods

Maximum Parsimony

Without the consideration of bayesian analysis, for any kind of data parsimonous methods are said to be the most efficient in retrieving the tree which is the closest to the traditional tree given by comparative method [64]. We first used this method to search for the most parsimonous tree from the given data. There are various types of parsimonies depending upon the number of states (binary or multi-state) and the kind of transitions between the states. In our study we limit ourselves to three kind

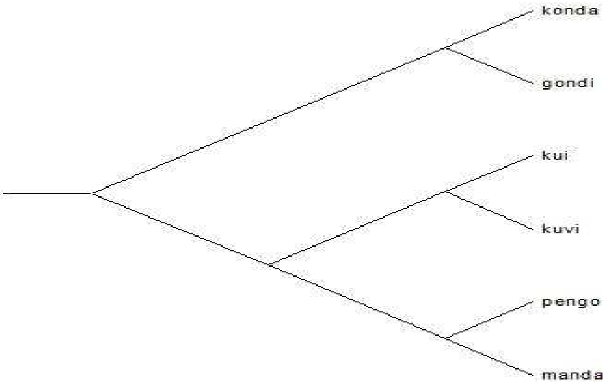


Figure 4.6: Phylogenetic tree using UPGMA

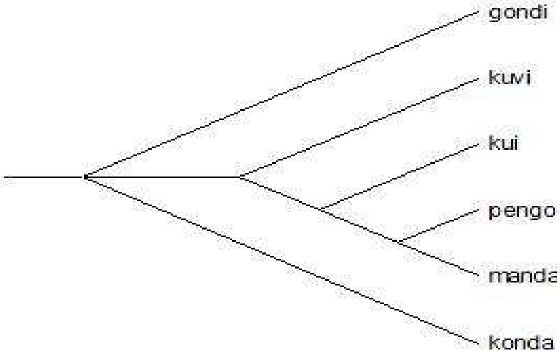


Figure 4.7: Phylogenetic tree using Neighbour Joining

of parsimonies Camin-Sokal, Wagner and Dollo parsimony. The assumptions of each method is given below [32].
Assumptions of Camin-Sokal and Wagner's parsimony

1. Ancestral states are known (Camin-Sokal) or unknown (Wagner).

2. Different characters evolve independently.

3. Different lineages evolve independently.

4. Changes 0 → 1 are much more probable than changes 1 → 0 (Camin-Sokal) or equally probable (Wagner).

5. Both of these kinds of changes are a priori improbable over the evolutionary time spans involved in the differentiation of the group in question.

6. Other kinds of evolutionary event such as retention of polymorphism are far less probable than 0 → 1 changes.

7. Rates of evolution in different lineages are sufficiently low that two changes in a long segment of the tree are far less probable than one change in a short segment.

The objections to some of these assumptions can be summarised in the following statements. The assumption that different lineages evolve independently is not justifiable since borrowing does occur between the lineages (In the case of lexical diffusion, the words are affected by the change in the other words in the lexicon. In our study, the lexical data which we used was carefully studied and any item with the slightest evidence of borrowing was discarded. Hence this need not be a concern in our case). We also tested the hypothesis of the sound change being irreversible by giving equal chance for the reversible direction. Camin-Soakal parsimony reflects the case of sound change being irreversible and Wagner parsimony allows for a equal probability for a sound change to be reversible.

Assumptions of Dollo's Parsimony

1. We know which state is the ancestral one (state 0).

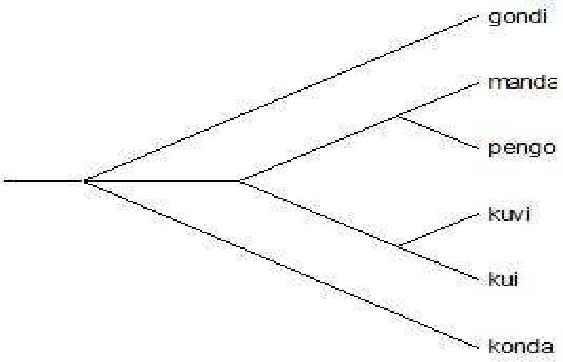


Figure 4.8: Phylogenetic tree using *PARS* method from *PHYLIP*

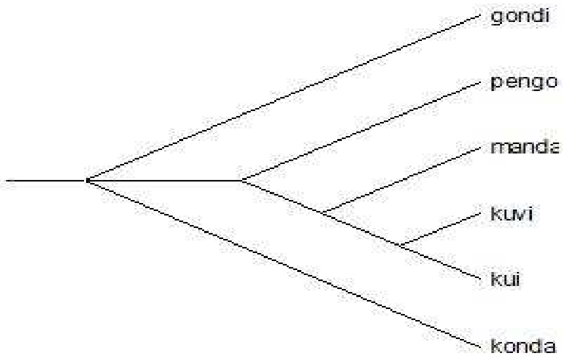


Figure 4.9: Phylogenetic tree using *PARS* method from *PHYLIP*

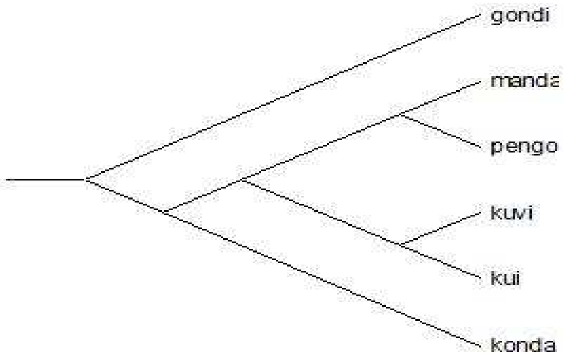


Figure 4.10: Phylogenetic tree using Camin-Soakal parsimony

2. The characters are evolving independently.

3. Different lineages evolve independently.

4. The probability of a forward change (0 → 1) is small over the evolutionary times involved.

5. The probability of a reversion (1 → 0) is also small, but still far larger than the probability of a forward change, so that many reversions are easier to envisage than even one extra forward change.

6. Retention of polymorphism for both states (0 and 1) is highly improbable.

7. The lengths of the segments of the true tree are not so unequal that two changes in a long segment are as probable as one in a short segment.

Dollo's parsimony is based on the law that traits can evolve only once. In this context, the evidence of cognates which represent the process of diffusion of sound change still in process, can be treated as trait. This is equivalent to stating that the sound change is homoplasy free. It has diffused over the languages in their common stage of evolution rather occuring at a later stage when the languages have diverged. This variety of parsimony also allows for determining the root of the tree.

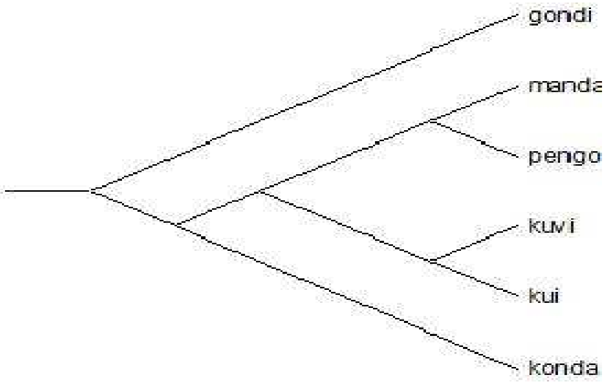


Figure 4.11: Phylogenetic tree using Dollo's parsimony

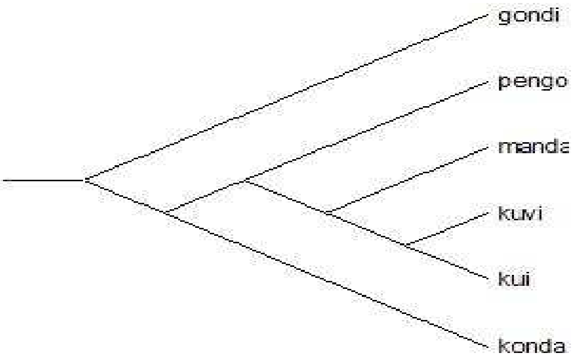


Figure 4.12: Phylogenetic tree using Dollo's parsimony

Bayesian Inference of Phylogenies

This is a recent class of methods which is an extension of maximum likelihood methods. We tried to use this method for inferring the tree from the character data. We used Metropolis-coupled Markov Chain Monte Carlo (MCMC) for sampling the posterior probabilities of the trees. The working of the method was explained in the Related Work section in detail. We would talk about the parameter settings and how we ran the experiments for inferring the tree. We tried using two priors a fixed shape parameter (α) and a uniform distribution. The results didnot vary much when we changed the priors. MCMC runs n chains out of which $n-1$ chains are heated. A heated chain has steady-state distribution $\pi_i(X) = \pi(X)^{\beta_i}$ with $\beta_i = \frac{1}{1+T(i-1)}$ where T is the temperature, i is the number of the chain and π is the posterior distribution and β is the power to which the posterior probability of each heated chain is raised to. The chains are heated in an incremental fashion and after each iteration, the states of two randomly picked chains i and j are swapped with the following probability

$$\min\left(1, \frac{\pi_i(X_t^{(j)})\pi_j(X_t^{(i)})}{\pi_i(X_t^{(i)})\pi_j(X_t^{(j)})}\right) \tag{4.2}$$

Inferences or sampling is usually done on the cold chain with $\beta = 1$ and $T = 0.20$ and the number of chains $n = 4$. We ran two independent analyses. The chains were kept running until the average deviation of the split frequencies between the two analyses was less than 0.01. The first 25% of the analyses were thrown out as the part of burn-in.

4.7 Discussion

We compare the results of all our experiments with the traditional tree topology given by Krishnamurti. To our surprise, UPGMA gives the tree which is the most consistent with the data given in table 4.1. In his 1983 paper Krishnamurti explains the issues present in the tree diagram 4.5. The tree makes 40 predictions out of which 37 are correct and 3 are wrong. The wrong predictions are 1) Kuvi should be closer to Konda than it is to Gondi but Kuvi shares 20 innovative items with Konda but 22 with Gondi 2) Konda should be closer to Manda than it is to Gondi but Konda shares 9 items with Manda but as many as 16 items with Gondi 3) Manda should be closer to Konda than it is to Gondi. The last prediction also turns out to be wrong since Manda shares 10 items with Gondi but only 9 items with Gondi. All of the above wrong predictions are rectified or donot appear in the tree given by UPGMA. By placing Gondi and Konda under the same subtree all the wrong predictions can be corrected. We donot comment about the other predictions because we are not aware of those at this moment. Interestingly, the neighbour joining method gives the same tree as the one obtained by Krishnamurti after they have applied their method on the data of two sound changes. Neighbour joining method returns an unrooted tree. So we rooted our tree using Gondi as a the outgroup and we obtained the rooted tree.

The results obtained in the next set of experiments using unchanged cognates as character-based data are very interesting. We use three variants of parsimony and each of them gives similar trees. Wagner's and Dollo's parsimonies return two most parsimonious trees whereas Carmin-Soakal's parsimony returns only one tree. The trees returned by Wagner's and Dollo's parsimonies are identical. All the parsimonious methods return the tree which is identical to comparative method. Wagner's and Dollo's return an extra tree. The tree returned by the method of Krishnamurti and Carmin-Soakal are the same. The extra tree returned by Wagner's and Dollo's is actually ranked second by Krishnamurti's method. This is actually an important result because the relaxation of the irreversibility of sound change constraint gives two trees with the same score[5]. In the case of Dollo's parsimony, the assumption is that change is very difficult to acquire but very easy to loose. This method also returns an extra tree which is ranked second by Krishnamurti.

After rigorously examining the method of Krishnamurti, we believe it to be a kind of parsimony with the same assumptions as Carmin-Soakal. We applied the Carmin-Soakal parsimony and scored the tree obtained by UPGMA and obtained a score of 79. In his analysis using single sound change Krishnamurti, considered only the trees which had a score ranging from 71 to 87 whose number was 45. Out of those 45 trees only the 11 lowest-scoring trees were considered. Their reason was that the trees with a score of 77 had Gondi and Konda reversed and disagrees with the lower scoring trees. We believe this solely cannot be the reason for not extending the study to other trees. As evident from the tree of figure 4.5, both the languages are not reversed but are grouped under the same subtree.

Examining the tree returned by bayesian analysis, we found that it returns essentially a tree identical to neighbour joining but with terenary branching with Gondi, Konda and the other languages as branches. The branch lengths returned by all the methods agree to the fact that Gondi has branched earlier than other languages which is followed by Konda. There is a general ambiguity about grouping of Manda and Pengo as well as Kui and Kuvi together.

[5] This is the case of Wagner's parsimony.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis we have tried to address two problems in historical linguistics namely Cognate Identification and Phylogenetic Trees. We have also tried to adress the problem of Letter to Phoneme Conversion which is very useful as a preprocessing step for Cognate Identification.

We have proposed two measures for identifying the cognates one based on distributional similarity, other based on feature n-gram DICE. The proposed method performs better than the earlier orthographic methods as it uses deeper phonetic information based on a rigorous mathematical model. The system was tested on a list of word pairs of length 250,000 out of which only 329 are genetic cognates. This shows the level of difficulty of the task of cognate identification. We evaluated our system against three baselines and we have achieved an improvement of 21%.

We have tried to address the problem of letter-to-phoneme conversion by modeling it as an SMT problem and we have used minimum error rate training to obtain the suitable model parameters, which according to our knowledge, is a novel approach to L2P task. We have experimented with minumum error rate training and the statistical machine translation toolkit Moses by representing every word as a sentence and every letter and phoeneme as a word. The results obtained are comparable to the state of the art system and our error analysis shows that a lot of improvement is still possible.

The trees we have obtained by using the unchanged cognates in south-central Dravidian language data as characters were very similar to the tree given by the comparative method. This is an attempt which has never been tried before. Unlike the work mentioned in section 4.1 which uses lexical, syntactic or morphological characters for inferring phylogenetic trees we use the cognates which are affected by the change as characters for determining the tree. All our attempts to root the tree using Gondi as the outgroup has yielded trees which concur to a large extent with the tree given by the comparative method. We also show that UPGMA performs better than neighbour joining in constructing the trees. Moreover, unlike the method proposed by Krishnamurti[1] the methods which we used are able to obtain the branch length of the tree. These branch lengths can be used to calibrate the divergence times of the tree and can throw light upon the antiquity of the Dravidian language family. This work reinforces the hypothesis that deeper linguistic features are more helpful in establishing the family tree than using lexical items for the same purpose.

5.2 Future work

All the work reported in the thesis can be extended in different directions. We mention some of the possible directions in which the work can be extended.

5.2.1 Possible Future Work on Cognate Identification

The performance of the cognate identification system can be improved by taking the sequence probabilities into consideration. We also propose a new measure which is actually a geometric mean of the precision of the various n-grams between the probability distributions of the word pair. One more aspect which can certainly improve the performance of the system is the weights given to the various articulatory features. By giving suitable weights to the articulatory features and designing a measure which takes the weights into consideration would probably increase the system's performance. One another aspect in the distributional similarity is the normalisation

[1] This work is based on his 1983 paper

factor. Whereas the orthographic measures are sequence based measures and are appropriately normalised by length, the symmetric cross entropy measure (SCE) has to be normalised by length. Finding the right way of normalisation would certainly improve the perfomance of the system. In this thesis we have only considered a single information theoretic measure i.e. SCE was used for measuring the distributional similarity. Testing with various other measures would be definitely a direction of research to follow.

5.2.2 Possible Future Work on Letter to Phoneme Conversion

Intuitively, the performance of the system can be improved in at least two areas. First is the Minimum Error Rate Training (MERT) and the second is the decoding phase. The MERT implementation currently uses the Bleu function [62] as the loss function. Bleu function calculates the geometric mean of the precision of n-grams of various lengths between the candidate and the reference translation. At present, the precision is calculated only up to four grams, which we believe is insufficient for the L2P task. This can be replaced with string similarity measures like Levenshtein distance or a 0-1 loss function or a combination of both. Incorporating more model parameters would help very much in improving the performance of the system.

Using phonetic feature based edit distance or string similarity as the loss function in the MERT implementation can improve results significantly. In addition, incorporating more model parameters and extensive testing of these parameters might improve the results of the system. We also plan to introduce a decoding scheme similar to the substring based transducer [72] to improve the usage of lower order language models.

5.2.3 Possible Future Work on Phylogenetic Trees

In this direction we intend to use the data with the second sound change for our experiments and observe whether we are able to improve the results than that of Krishnamurti [52]. Another direction for this work is to use the penalised likelihood methods for estimating the divergence times for the various trees. Although some work was done in the past for Dravidian languages using Swadesh list [5], the rise of new techniques in computational biology has reopened the issue whether preparing the Swadesh list can answer many of the open challenges in Dravidian language family. We also intend to use the same methods to determine whether there was a terenary or a binary split in the Dravidian family. For this we intend to use the morpho-syntactic and phonological data presented in the current edition of Dravidian Languages [51]. Also, not in the near future, we wish to prepare a Swadesh list for Dravidian languages and apply the above methods for dating the nodes in the family tree.

Appendix A

Phylogenetic trees for a linguistic area

A.1 Introduction

Establishing relationships among languages which have been in contact for a long time has been a topic of interest in historical linguistics [19]. However, this topic has been much less explored in the computational linguistics community. Most of the previous work is focused on reconstruction of phylogenetic trees for a particular language family using handcrafted word lists [34, 7, 9, 58] or using synthetic data [11].

In this paper we pose the following questions. What happens when we try to construct phylogenetic trees using inter-language distances in the context of a *linguistic area*[1]? Can the phylogenetic trees be used for evaluating the robustness of the inter-language distance measures and the meaningfulness of the distances? To our knowledge these questions have not been addressed previously. As Singh and Surana [74] showed, corpus based measures can be successfully used for comparative study of languages. Can these distances, estimated from a noisy corpus[2], meaningfully be used to construct phylogenetic trees? Can the information represented by

[1] The term *linguistic area* or *Sprachbund* [29] refers to a group of languages that have become similar in some way as a result of proximity and *language contact*, even if they belong to different families. The best known example is the Indian (or South Asian) linguistic area.

[2] By noisy corpus we mean a corpus that includes wrongly spelled words and spelling variations.

the tree give meaningful interpretations about the languages involved? In this paper, we try to answer these questions. By using meaningful measures for estimating the distance between languages, we try to establish that the answers to these questions are affirmative. Overall, the contributions of the paper are the following a) use a new measure for estimating language distance b) present results of the experiments on constructing phylogenetic trees from corpus based word lists rather than handcrafted ones c) validate the hypothesis that India is a linguistic area [29].

The paper is organized as follows. Related work is discussed in Section 2. A brief discussion of various inter-language measures is given in Section 3. The experimental setup and the analysis of the results have been given in Section 4 and Section 5, respectively. We present a summary of our experiments, analysis of the results and future directions of the work in Section 6.

A.2 Related Work

In recent years, the methods developed in computational biology [35, 68, 31, 80] have been successfully adapted in computational linguistics for constructing the phylogeny[3]. All these methods are character based or distance based methods. The major disadvantage of these approaches is that they require handcrafted lists. Moreover, the methods inspired from glottochronology take a boolean matrix as input, which denotes the change in the state of the 'characters' (the 'characters' can be lexical, morphological or phonological) to infer the phylogenetic trees.

Ellison and Kirby [28] discuss establishing a probability distribution for every language through intra-lexical comparison using confusion probabilities. They use normalized edit distance to calculate the probabilities. Then the distance between every language pair is estimated as a distance between the probability distributions formed for individual languages. The distances (between languages) are estimated using KL-divergence and Rao's distance. The same measures are also used to find

[3] Phylogeny is the (study of) evolutionary development and history of a species or higher taxonomic grouping of organisms. The term is now also used for other things such as tribes and languages. Phylogenetic trees represent this evolutionary development.

the level of cognacy between the words. The experiments are conducted on Dyen's [27] classical Indo-European dataset. The estimated distances are used for constructing a phylogenetic tree of the Indo-European languages.

Bouchard-Cote et al. [16], in a novel attempt, combine the advantages of classical comparative method and the corpus-based probabilistic models. The word forms are represented by phoneme sequences which undergo stochastic edits along the branches of a phylogenetic tree. The robustness of the model is proved when it selects the linguistically attested phylogeny. The stochastic models successfully model the language change by using synchronic languages to reconstruct the word forms in Vulgar Latin and Classical Latin. Although it reconstructs the ancient word forms of the Romance Languages, a major disadvantage of this model is that some amount of data of the ancient word forms is required to train the model, which may not be available in many cases.

In another novel attempt, Singh and Surana [74] used corpus based simple measures to show that corpus can be used for comparative study of languages. They used both character n-gram distances and Surface Similarity [75] to identify the potential cognates[4], which in turn are being used to estimate the inter-language distance. Both diachronic and synchronic experiments are performed and the results very well attest to the linguistic facts. They also argued that there is a common orthographic as well as phonetic space for languages with a long history of contact which can be exploited for developing inter-language (rather than intra-language) measures, in contrast to the position taken by Ellison and Kirby [28]. Having followed this line of argument, we explain some corpus measures which we adopted from their work and also use a new measure which we call phonetic (and orthographic) feature n-gram based distance.

[4] Potential cognates are words of different languages which are similar in form and therefore are likely to be cognates. They might include some 'false friends', i.e., words which are not etymologically inherited. It is worthwhile to experiment (using statistical techniques) on potential cognates, even without removing the 'false friends' because a large percentage of them are actually cognates in the linguistic sense.

A.3 Inter-Language Measures

Such measures can be broadly divided into three categories. Character n-gram measures, cognate based measures and feature n-gram measures. The following sections describe each measure in more detail. One important point that can be mentioned here is that all the languages we experimented on use Brahmi origin scripts, which have almost one-to-one correspondence between letters and phonemes. Moreover, these scripts are similar in a lot of ways, especially the fact that the alphabets used by them can be seen as subsets of the same abstract alphabet, although the letters may have different shapes so that to a lay person the scripts seem very different. In fact, there is a 'super encoding' or 'meta encoding' called ISCII that can be used to represent this common alphabet. The letters of this common alphbet can be approximately treated like phonemes for computational purposes. For languages which do not use such scripts, we will first have to convert the text into a phonetic notation to be able to use the methods described below, except perhaps the first one.

A.3.1 Symmetric Cross Entropy (SCE)

The first measure is purely a letter n-gram based measure similar to the one used by Singh [76] for language and encoding identification. Note that since letters in Brahmi origin scripts can almost be treated like phonemes, we could call this method a phoneme n-gram based measure. To calculate the distance, letter 5-gram models are prepared from the corpora of the languages to be compared. Then the n-grams of all sizes (unigrams, bigrams, etc.) are combined and sorted according to their probability in descending order. Only the top N n-grams are retained and the rest are pruned. This is based on the results obtained by Cavnar [20] and validated by Singh, which show that the top N (300 according to Cavnar) n-grams have a high correlation with the identity of the language. At this stage there are two probability distributions which can be compared by a measure of distributional similarity. The measure used here is symmetric cross entropy:

$$d_{sce} = \sum_{g_l = g_m} (p(g_l)\ log\ q(g_m) + q(g_m)\ log\ p(g_l)) \tag{A.1}$$

where p and q are the probability distributions for the two languages and g_l and g_m are n-grams in languages l and m, respectively. The probabilities of bigrams and larger n-grams are relative frequencies over a single distribution consisting of n-grams of all sizes up to 5 (the 'order' of the n-gram model), not conditional probabilities, as in standard n-gram models for calculating sequence probabilities.

The disadvantage of this measure is that it does not use any linguistic (e.g., phonetic) information, but the advantage is that it can easily measure the similarity of distributions of n-grams. Such measures have proved to be very effective in automatically identifying languages of text, with accuracies nearing 100% for fairly small amounts of training and test data [2, 76].

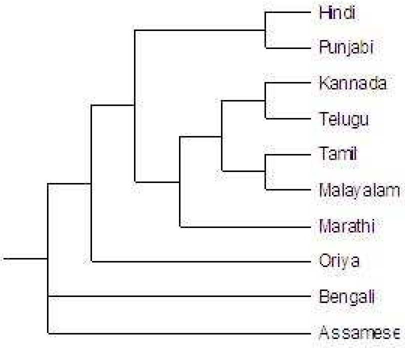


Figure A.1: Phylogenetic tree using SCE

A.3.2 Measures based on Cognate Identification

The other two measures are based on potential cognates, i.e., words of similar form. Both of them use an algorithm for identification of potential cognates. Many such algorithms have been proposed. For identifying cognates, Singh and Surana [74] used the Computational Phonetic Model of Scripts or CPMS [75]. This model takes into account the characteristics of Brahmi origin scripts and calculates Surface Similarity. It consists of a model of alphabet that represents the common alphabet for Brahmi origin scripts, a model of phonology that maps the letters (which are, for the most part, phonemes) to phonetic and orthographic features, a Stepped Distance Function (SDF) that calculates the phonetic and orthographic similarity of two letters and a dynamic programming (DP) algorithm that calculates the Surface Similarity of two words or strings. The CPMS was adapted by Singh and Surana for identifying the potential cognates.

In general, the distance between two strings can be defined as:

$$c_{lm} \ = f_p(w_l, \ w_m) \tag{A.2}$$

where f_p is the function (implemented as a DP alignment algorithm) which calculates Surface Similarity using the CPMS based cost between the word w_l of language l and the word w_m of language m.

Those word pairs are identified as cognates which have the least cost.

Cognate Coverage Distance (CCD)

The second measure used is a corpus based estimate of the coverage of cognates across two languages. Cognate coverage is defined ideally as the number of words (from the vocabularies of the two languages) which are of the same origin, but which is approximately estimated by identifying words of similar form (potential cognates). The decision about whether two words are cognates or not is made on the basis of Surface Similarity of the two words as described in the previous section. Non-parallel corpora of the two languages are used for identifying the cognates.

The normalized distance between two languages is defined as:

$$t'_{lm} \ = \ 1 - \frac{t_{lm}}{max(t)} \tag{A.3}$$

where t_{lm} and t_{ml} are the number of (potential) cognates found when comparing from language l to m and from language m to l, respectively.

Since the CPMS based measure of Surface Similarity is asymmetric, the average number of unidirectional cognates is calculated:

$$d^{ccd} = \frac{t'_{lm} + t'_{ml}}{2} \tag{A.4}$$

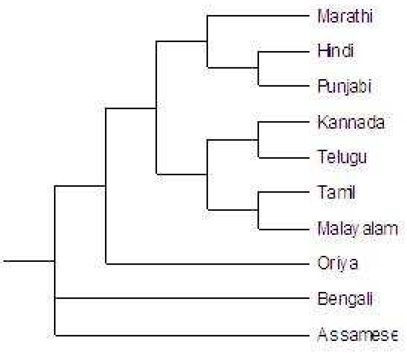


Figure A.2: Phylogenetic tree using CCD

Phonetic Distance of Cognates (PDC)

Simply finding the coverage of cognates may indicate the distance between two languages, but a measure based solely on this information does not take into account the variation between the cognates themselves. To include this variation into the estimate of distance, Singh and Surana [74] used another measure based on the sum of the CPMS based cost of n cognates found between two languages:

$$C^{pdc}_{lm} = \sum_{i=0}^{n} c_{lm} \tag{A.5}$$

where n is the minimum of t_{lm} for all the language pairs compared.

The normalized distance can be defined as:

$$C'_{lm} = \frac{C^{pdc}_{lm}}{max(C^{pdc})} \tag{A.6}$$

A symmetric version of this cost is then calculated:

$$d_{pdc} = \frac{C'_{lm} + C'_{ml}}{2} \tag{A.7}$$

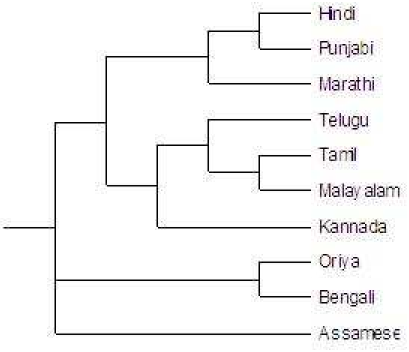


Figure A.3: Phylogenetic tree using PDC

A.3.3 Feature N-Grams (FNG)

The idea in using this measure is that the way phonemes occur together matters less than the way the phonetic features occur together because phonemes themselves are defined in terms of the features. Therefore, it makes more sense to a have measure directly in terms of phonetic features. But since we are experimenting directly with corpus data (without any phonetic transcription) using the CPMS [75], we also include some orthographic features as given in the CPMS implementation. The letter to feature mapping that we use comes from the CPMS. Basically, each word is converted into a set of sequences of feature-value pairs such that any feature can follow any feature, which means that the number of sequences for a word of length l_w is less than or equal to $(N_f \times N_v)^{l_w}$, where N_f is the number of possible features and N_v is the number of possible values. We create sequences of feature-value pairs for all the words and from this 'corpus' of feature-value pair sequences we build the feature n-gram model.

The formula for calculating distributional similarity based on these phonetic and orthographic features is the same (SCE) as given in equation 1, except that the distribution in this case is made up of features rather than letters. Note that since we do not assume the features to be independent, any feature can follow any other feature in a feature n-gram. All the permutations are computed before the feature n-gram model is pruned to keep only the top N feature n-grams. The order of the n-gram model is kept as 3, i.e., trigrams.

The feature n-grams are computed as follows. For a given word, each letter is first converted into a vector consisting of the feature-value pairs which are mapped to it by the CPMS. Then, from the sequence of vectors of features, all possible sequences of features up to the length 3 (the order of the n-gram model) are computed. All these sequences of features (feature n-grams) are added to the n-gram model. Finally the model is pruned as mentioned above. We expected this measure to work better because it works at a higher level of abstraction and is more linguistically valid.

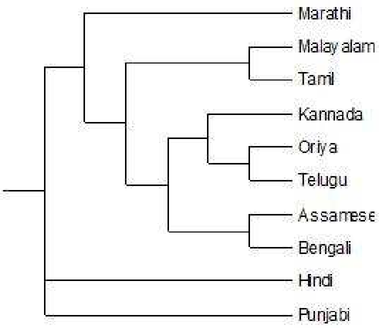


Figure A.4: Phylogenetic tree using feature n-grams

A.4 Experimental Setup

Although the languages we selected belong to two different language families, there are a lot of similarities among them which allow us to choose them for our experiments [29]. The corpora used for our experiments are all part of the CIIL multilingual corpus. The experiments were conducted using word lists prepared from the raw corpus for every language. No morph analyzer or stemmer has been applied to the words. Initially the word types with their frequencies are extracted from the corpus. Then the word types are sorted based on their corresponding frequency. Only the top N_w of these word types are retained. This is done with the aim of including as much of the core vocabulary as possible for comparing the languages[5]. For using cognate

[5] For our experiments we fixed N_w at 50,000. This number is different from N, the number of top n-grams that are retained after pruning the n-gram model.

based measures for estimation of language distance, cognates are extracted from the word lists between these languages. For feature n-gram measures, the feature n-gram models are prepared as explained in Section 3.

We calculate the distance between every pair of languages available. We compare the results between all the four measures discussed above by constructing trees using these measures. The trees are constructed using the NEIGHBOR program in the PHYLIP package[6]. The NEIGHBOR programs provides two distance-based tree construction algorithms: Neighbour Joining and UPGMA. For our experiments we used Neighbour Joining as it does not assume a constant rate of evolution and it produces unrooted trees unlike UPGMA which assumes constant rate of evolution (the length of the leaves from the root of the tree is same across all the leaves) and produces rooted trees. We do not do any outgrouping as outgrouping makes sense only when all the languages belong to a single family.

A.5 Analysis of Results

Table 1 shows the results obtained for the four distance measures. Figures 1 to 4 show the trees obtained using all the above measures. There are three subgroupings of the languages which are clearly visible in all the trees. Namely, Northern Indo-Aryan (Hindi and Punjabi), Eastern Indo-Aryan (Assamese, Bengali and Oriya) and Dravidian languages (Tamil, Kannada, Malayalam and Telugu). There are clearly some similarities in the trees which are generated by all the methods. All the methods group Hindi and Punjabi, Tamil and Malayalam together. CCD gives the normalized measure of the number of cognates between every language pair. In the case of CCD tree, although Bengali and Assamese are grouped together, Oriya is placed incorrectly, which is correctly placed in the case of feature n-grams.

Oriya is incorrectly grouped with Bengali in the case of PDC tree. The reason can be because of the huge number of shared words which cause a lower phonetic distance between the languages. Kannada and Telugu are not grouped together in the case of PDC. Marathi is either classified with Northern Indo-Aryan languages or

[6] http://evolution.genetics.washington.edu/phylip/phylip.html

with Dravidian languages. It is grouped with Indo-Aryan languages in the case of cognate distance measures and grouped with Dravidian languages in the other cases. The reason for grouping it with Dravidian languages is the influence of Dravidian languages due to long history of contact.

The distance of a terminal node from its parent gives very important information[7]. For example, Tamil is always at a greater distance from its parent node, although grouped with Malayalam, compared to other languages. Especially in the case of feature n-grams and SCE, the distance is very evident. The reason for this is the lower number of 'characters' (elements from which n-grams are made) when compared to other languages in the case of SCE. In the case of feature n-grams, the lack of phonemic distinction in writing between voiced and unvoiced sounds for Tamil decreases the number of shared feature n-grams. Moreover, the number of borrowings from Indo-Aryan Languages are comparatively less in the case of Tamil.

[7] The trees in the figures are not scaled, but the distances are given in the table.

	BN	**HI**	**KN**	**ML**	**MR**	**OR**	**PA**	**TA**	**TE**
AS	0.02	0.39	0.71	0.86	0.61	0.20	0.61	0.93	0.73
	0.12	0.25	0.39	0.61	0.45	0.11	0.58	0.95	0.46
	0.05	0.30	0.51	0.50	0.43	0.18	0.42	0.70	0.64
	0.02	0.06	0.07	0.12	0.09	0.05	0.09	0.13	0.05
BN		0.32	0.68	0.86	0.57	0.07	0.56	0.96	0.70
		0.29	0.42	0.64	0.42	0.05	0.56	0.90	0.50
		0.29	0.47	0.45	0.43	0.14	0.42	0.74	0.43
		0.06	0.07	0.13	0.08	0.04	0.09	0.11	0.02
HI			0.61	0.81	0.42	0.40	0.20	0.93	0.61
			0.17	0.56	0.16	0.27	0.16	0.87	0.38
			0.43	0.46	0.16	0.33	0.20	0.74	0.34
			0.09	0.09	0.06	0.08	0.03	0.15	0.13
KN				0.77	0.68	0.75	0.73	0.88	0.53
				0.45	0.17	0.31	0.50	0.82	0.25
				0.18	0.38	0.52	0.58	0.42	0.09
				0.10	0.09	0.02	0.08	0.10	0.03
ML					0.89	0.88	0.88	0.62	0.72
					0.65	0.59	0.77	0.56	0.31
					0.42	0.53	0.55	0.07	0.19
					0.13	0.13	0.11	0.07	0.15
MR						0.64	0.52	0.95	0.68
						0.40	0.37	0.94	0.46
						0.34	0.39	0.60	0.30
						0.08	0.06	0.13	0.09
OR							0.63	0.98	0.74
							0.45	0.89	0.44
							0.65	0.83	0.64
							0.07	0.10	0.00
PA								0.90	0.71
								0.90	0.59
								0.92	0.48
								0.14	0.07
TA									0.85
									0.81
									0.39
									0.08

AS: Assamese, **BN:** Bengali, **HI:** Hindi, **KN:** Kannada
ML: Malayalam, **MR:** Marathi, **OR:** Oriya,
PA: Punjabi, **TA:** Tamil, **TE:** Telugu

Table A.1: Inter-language comparison among ten major South Asian languages using four corpus based measures. The values have been normalized and scaled to be somewhat comparable. Each cell contains four values: by CCD, PDC, SCE and FNG.

Appendix B

Machine Transliteration as a SMT Problem

B.1 Introduction

Transliteration can be defined as the task of transcribing the words from a source script to a target script [78]. Transliteration systems find wide applications in Cross Lingual Information Retrieval Systems (CLIR) and Machine Translation (MT) systems. The systems also find use in sentence aligners and word aligners [6]. Transcribing the words from one language to another language without the use of a bilingual lexicon is a challenging task as the output word produced in target language should be such that it is acceptable to the readers of the target language. The difficulty arises due to the huge number of Out Of Vocabulary (OOV) words which are continuously added into the language. These OOV words include named entities, technical words, borrowed words and loan words.

In this paper we present a technique for transliterating named entities from English to Hindi using a small set of training and development data. The paper is organised as follows. A survey of the previous work is presented in the next subsection. Section 2 describes the problem modeling which we have adopted from [63] which they use for L2P task. Section 3 describes how the parameters are tuned for optimal performance. A brief description of the data sets is provided in Section 4. Section 5 has the results which we have obtained for the test data. Finally we conclude with a summary of the methods and a analysis of the errors.

B.1.1 Previous Work

Surana and Singh [78] propose a transliteration system in which they use two different ways of transliterating the named entities based on their origin. A word is classified into two classes either Indian or foreign using character based n-grams. They report their results on Telugu and Hindi data sets. Sherif and Kondrak [71] propose a hybrid approach in which they use the Veterbi-based monotone search algorithm for searching the possible candidate transliterations. Using the approach given in [66] the sub-string translations are learnt. They integrate the word-based unigram model based on [39, 4] with the above model for improving the quality of transliterations.

Malik et al [53] try to solve a special case of transliteration for Punjabi in which they convert from Shahmukhi (Arabic script) to Gurumukhi using a set of transliteration rules. Abdul Jaleel et al [1] show that, in the domain of information retrieval, the cross language retrieval performance was reduced by 50% when the name entities were not transliterated.

B.2 Problem Modeling

Assume that given a word, represented as a sequence of letters of the source language $\mathbf{s} = s_1^J = s_1...s_j...s_J$, needs to be transcribed as a sequence of letters in the target language, represented as $\mathbf{t} = t_1^I = t_1...t_i...t_I$. The problem of finding the best target language letter sequence among the transliterated candidates can be represented as:

$$\mathbf{t}_{best} = \arg\max_{\mathbf{t}} \{\Pr(\mathbf{t} \mid \mathbf{s})\} \tag{B.1}$$

We model the transliteration problem based on the noisy channel model. Reformulating the above equation using Bayes Rule:

$$\mathbf{t}_{best} = \arg\max_{\mathbf{t}} p(\mathbf{s} \mid \mathbf{t})\, p(\mathbf{s}) \tag{B.2}$$

This formulation allows for a target language letters' n-gram model $p\,(\mathbf{t})$ and a transcription model $p\,(\mathbf{s} \mid \mathbf{t})$. Given a sequence of letters $\mathbf{s}$, the argmax function is a search function to output the best target letter sequence.

From the above equation, the best target sequence is obtained based on the product of the probabilities of transcription model and the probabilities of a language model and their respective weights. The method for obtaining the transcription probabilities is described briefly in the next section. Determining the best weights is necessary for obtaining the right target language sequence. The estimation of the models' weights can be done in the following manner.

The posterior probability $\Pr\,(\mathbf{t} \mid \mathbf{s})$ can also be directly modeled using a log-linear model. In this model, we have a set of M feature functions $h_m(\mathbf{t},\, \mathbf{s}), m = 1...M$. For each feature function there exists a weight or model parameter $\lambda_m, m = 1...M$. Thus the posterior probability becomes:

$$\Pr\,(\mathbf{t} \mid \mathbf{s}) = p_{\lambda_1^M}\,(\mathbf{t} \mid \mathbf{s}) \tag{B.3}$$

$$= \frac{\exp\left[\Sigma_{m=1}^{M}\lambda_m h_m(\mathbf{t},\, \mathbf{s})\right]}{\sum_{\acute{t}_1^I}\exp\left[\Sigma_{m=1}^{M}\lambda_m h_m(\acute{t}_1^I, \mathbf{s})\right]} \tag{B.4}$$

with the denominator, a normalization factor that can be ignored in the maximization process.

The above modeling entails finding the suitable model parameters or weights which reflect the properties of our task. We adopt the criterion followed in [60] for optimising the parameters of the model. The details of the solution and proof for the convergence are given in [60]. The models' weights, used for the transliteration task, are obtained from this training.

All the above tools are available as a part of publicly available MOSES [40] tool kit. Hence we used the tool kit for our experiments.

B.3 Tuning the parameters

The source language to target language letters are aligned using GIZA++ [61]. Every letter is treated as a single word for the GIZA++ input. The alignments are then used to learn the phrase transliteration probabilities which are estimated using the scoring function given in [42].

The parameters which have a major influence on the performance of a phrase-based SMT model are the alignment heuristics, the maximum phrase length (MPR) and the order of the language model [42]. In the context of transliteration, *phrase* means a sequence of letters(of source and target language) mapped to each other with some probability (i.e., the *hypothesis*) and stored in a phrase table. The *maximum phrase length* corresponds to the maximum number of letters that a hypothesis can contain. Higher phrase length corresponds a larger phrase table during decoding.

We have conducted experiments to see which combination gives the best output. We initially trained the model with various parameters on the training data and tested for various values of the above parameters. We varied the maximum phrase length from 2 to 7. The language model was trained using SRILM toolkit [77]. We varied the order of language model from 2 to 8. We also traversed the alignment heuristics spectrum, from the parsimonious *intersect* at one end of the spectrum through *grow, grow-diag, grow-diag-final, grow-diag-final-and* and *srctotgt* to the most lenient *union* at the other end.

We observed that the best results were obtained when the language model was trained on 7-gram and the alignment heuristic was *grow-diag-final.* No significant improvement was observed in the results when the value of MPR was greater than 7. We have taken care such that the alignments are always monotonic and no letter was left unlinked.

B.4 Data Sets

Prior to the release of the test data only the training data and development data was available. The training data and development data consisted of a parallel corpus having entries in both English and Hindi. The training data and development data had 9975 entries and 974 entries. We used the training data given as a part of the shared task for generating the phrase table and the language model. For tuning the parameters mentioned in the previous section, we used the development data.

From the training and development data we have observed that the words can be roughly divided into following categories, Persian, European (primarily English), Indian, Arabic words, based on their origin. The test data consisted of 1000 entries. We proceeded to experiment with the test set once the set was released.

B.5 Experiments and Results

The parameters described in Section 3 were the initial settings of the system. The system was tuned on the development set, as described in Section 2, for obtaining the appropriate model weights. The system tuned on the development data was used to test it against the test data set. We have obtained the following model weights.

```
language model = 0.099
translation model = 0.122
```

Prior to the release of the test data, we tested the system without tuning on development data. The default model weights were used to test our system on the development data. In the next step the model weights were obtained by tuning the system. Although the system allows for a distortion model, allowing for phrase movements, we did not use the distortion model as distortion is meaningless in the domain of transliteration. The following measures were used to evaluate our system performance. Word Accuracy (ACC), Mean F-Score, Mean Reciprocal Rank (MRR), MAP_{ref}, MAP_{10}, MAP_{sys}. A detailed description of each measure is available in [1].

[1] https://translit.i2r.a-star.edu.sg/news2009/whitepaper/

Measure	Result
ACC	0.463
Mean F-Score	0.876
MRR	0.573
MAP_{ref}	0.454
MAP_{10}	0.201
MAP_{sys}	0.201

Table B.1: Evaluation of Various Measures on Test Data

B.6 Conclusion

In this paper we show that we can use the popular phrase based SMT systems successfully for the task of transliteration. The publicly available tool GIZA++ was used to align the letters. Then the phrases were extracted and counted and stored in phrase tables. The weights were estimated using minimum error rate training as described earlier using development data. Then A* based decoder was used to transliterate the English words into Hindi. After the release of the reference corpora we examined the error results and observed that majority of the errors resulted in the case of the foreign origin words.

Bibliography

[1] N. AbdulJaleel and L.S. Larkey. Statistical transliteration for english-arabic cross language information retrieval. 2003.

[2] G. Adams and P. Resnik. A Language Identification Application Built on the Java Client/Server Platform. *From Research to Commercial Applications: Making NLP Work in Practice*, pages 43–47, 1997.

[3] Y. Al-Onaizan, J. Curin, M. Jahr, K. Knight, J. Lafferty, D. Melamed, F.J. Och, D. Purdy, N.A. Smith, and D. Yarowsky. Statistical machine translation. In *Final Report, JHU Summer Workshop*, 1999.

[4] Y. Al-Onaizan and K. Knight. Machine transliteration of names in Arabic text. In *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*, pages 1–13. Association for Computational Linguistics Morristown, NJ, USA, 2002.

[5] M. Andronov. Lexicostatistic analysis of the chronology of disintegration of proto-Dravidian. *Indo-Iranian Journal*, 7(2):170–186, 1964.

[6] N. Aswani and R. Gaizauskas. A hybrid approach to align sentences and words in English-Hindi parallel corpora. *Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, page 57, 2005.

[7] Q. Atkinson, G. Nicholls, D. Welch, and R. Gray. From words to dates: water into wine, mathemagic or phylogenetic inference? *Transactions of the Philological Society*, 103(2):193–219, 2005.

[8] Q.D. Atkinson and R.D. Gray. Curious parallels and curious connections: Phylogenetic thinking in biology and historical linguistics. *Systematic Biology*, pages 513–526, 2005.

[9] QD Atkinson and RD Gray. How old is the Indo-European language family? Progress or more moths to the flame. *Phylogenetic Methods and the Prehistory of Languages (Forster P, Renfrew C, eds)*, pages 91–109, 2006.

[10] D. Bakker. LINFER and the WALS database. In *Workshop on Interpreting Typological Distributions, Leipzig*, 2004.

[11] F. Barbançon, T. Warnow, S.N. Evans, D. Ringe, and L. Nakhleh. An experimental study comparing linguistic phylogenetic reconstruction methods. Technical report, Technical Report 732, Department of Statistics, University of California, Berkeley, 2007.

[12] Susan Bartlett, Grzegorz Kondrak, and Colin Cherry. Automatic syllabification with structured SVMs for letter-to-phoneme conversion. In *Proceedings of ACL-08: HLT*, pages 568–576, Columbus, Ohio, June 2008. ACL.

[13] Shane Bergsma and Grzegorz Kondrak. Alignment-based discriminative string similarity. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 656–663, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[14] Max Bisani and Hermann Ney. Investigations on joint-multigram models for grapheme-to-phoneme conversion. In *International Conference on Spoken Language Processing*, pages 105–108, Denver, CO, USA, September 2002.

[15] A.W. Black, K. Lenzo, and V. Pagel. Issues in Building General Letter to Sound Rules. In *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*. ISCA, 1998.

[16] A. Bouchard-Cote, P. Liang, T.L. Griffiths, and D. Klein. A probabilistic approach to language change. NIPS, 2000.

[17] A. Bouchard-Cote, P. Liang, T.L. Griffiths, and D. Klein. A Probabilistic Approach to Diachronic Phonology. *Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL)*, 2007.

[18] A. Bouchard-Cote, P. Liang, T.L. Griffiths, and D. Klein. A probabilistic approach to language change. EMNLP, 2007.

[19] L. Campbell. *Historical linguistics: an introduction.* MIT Press, 2004.

[20] W.B. Cavnar and J.M. Trenkle. N-gram-based text categorization. *Ann Arbor MI*, 48113:4001, 1994.

[21] K.W. Church. Char_align: a program for aligning parallel texts at the character level. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 1–8. Association for Computational Linguistics Morristown, NJ, USA, 1993.

[22] M. Collins. Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on EMNLP-Volume 10*, pages 1–8. ACL, Morristown, NJ, USA, 2002.

[23] K. Crammer and Y. Singer. Ultraconservative online algorithms for multiclass problems. *The Journal of Machine Learning Research*, 3:951–991, 2003.

[24] Walter M. P. Daelemans and Antal P. J. van den Bosch. Language-Independent Data-0riented Grapheme-to-Phoneme Conversion. *Progress in Speech Synthesis*, 1997.

[25] R.I. Damper, Y. Marchand, J.D. Marseters, and A. Bazin. Aligning Letters and Phonemes for Speech Synthesis. In *Fifth ISCA Workshop on Speech Synthesis.* ISCA, 2004.

[26] R.G. D'Andrade. U-statistic hierarchical clustering. *Psychometrika*, 43(1):59–67, 1978.

[27] I. Dyen, J.B. Kruskal, and P. Black. An Indoeuropean classification: a lexicostatistical experiment. Amer Philosophical Society, 1992.

[28] T.M. Ellison and S. Kirby. Measuring language divergence by intra-lexical comparison. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 273–280. Association for Computational Linguistics Morristown, NJ, USA, 2006.

[29] MB Emeneau. India as a Lingustic Area. *Language*, pages 3–16, 1956.

[30] S.N. Evans, D. Ringe, and T. Warnow. Inference of divergence times as a statistical inverse problem. *Phylogenetic Methods and the Prehistory of Languages. McDonald Institute Monographs*, pages 119–130, 2004.

[31] J. Felsenstein. Inferring Phylogenies. Sunderland, MA. *Sinauer Press. Chapters*, 1(7):11, 2003.

[32] J. Felsenstein and J. Felenstein. *Inferring phylogenies*. Sinauer Associates Sunderland, Mass., USA, 2003.

[33] O. Frunza and D. Inkpen. Semi-supervised learning of partial cognates using bilingual bootstrapping. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 441–448. Association for Computational Linguistics Morristown, NJ, USA, 2006.

[34] R.D. Gray and Q.D. Atkinson. Language-tree divergence times support the Anatolian theory of Indo-European origin. *Earth Planet. Sci*, 23:41–63, 1995.

[35] J.P. Huelsenbeck, F. Ronquist, R. Nielsen, and J.P. Bollback. Bayesian inference of phylogeny and its impact on evolutionary biology. *Science*, 294(5550):2310–2314, 2001.

[36] D. Inkpen, O. Frunza, and G. Kondrak. Automatic identification of cognates and false friends in french and english. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 251–257, 2005.

[37] Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. Joint processing and discriminative training for letter-to-phoneme conversion. In *Proceedings of ACL-08: HLT*, pages 905–913, Columbus, Ohio, June 2008. ACL.

[38] Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *HLT 2007: The Conference of the NAACL; Proceedings of the Main Conference*, pages 372–379, Rochester, New York, April 2007. ACL.

[39] K. Knight and J. Graehl. Machine transliteration. *Computational Linguistics*, 24(4):599–612, 1998.

[40] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL*, volume 45, page 2, 2007.

[41] P. Koehn and K. Knight. Learning a translation lexicon from monolingual corpora. In *Proceedings of ACL Workshop on Unsupervised Lexical Acquisition*, volume 34, 2002.

[42] P. Koehn, F.J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the NAACL:HLT-Volume 1*, pages 48–54. ACL Morristown, NJ, USA, 2003.

[43] J. Kominek and A.W. Black. Learning pronunciation dictionaries: language complexity and word selection strategies. In *HLT-NAACL*, pages 232–239. ACL, Morristown, NJ, USA, 2006.

[44] G. Kondrak. A new algorithm for the alignment of phonetic sequences. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 288–295, 2000.

[45] G. Kondrak. Identifying cognates by phonetic and semantic similarity. In *North American Chapter Of The Association For Computational Linguistics*, pages 1–8. Association for Computational Linguistics Morristown, NJ, USA, 2001.

[46] G. Kondrak. *Algorithms for language reconstruction.* University of Toronto Toronto, Ont., Canada, Canada, 2002.

[47] G. Kondrak. Combining evidence in cognate identification. *Lecture notes in computer science*, pages 44–59, 2004.

[48] G. Kondrak. Cognates and word alignment in bitexts. In *Proceedings of the 10th Machine Translation Summit*, pages 305–312, 2005.

[49] G. Kondrak and T. Sherif. Evaluation of several phonetic similarity algorithms on the task of cognate identification. In *Proceedings of the Workshop on Linguistic Distances*, pages 43–50, 2006.

[50] B. Krishnamurti. Areal and lexical diffusion of sound change. *Language*, 54(1):1–20, 1978.

[51] B. Krishnamurti. *The Dravidian languages.* Cambridge University Press, 2003.

[52] B. Krishnamurti, L. Moses, and D. Danforth. Unchanged cognates as a criterion in linguistic subgrouping. *Language*, 59(3):541–568, 1983.

[53] M.G.A. Malik. Punjabi machine transliteration. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 1137–1144. Association for Computational Linguistics Morristown, NJ, USA, 2006.

[54] C.D. Manning and H. Schutze. *Foundations of statistical natural language processing.* MIT Press, 1999.

[55] Y. Marchand and R.I. Damper. A Multistrategy Approach to Improving Pronunciation by Analogy. *Computational Linguistics*, 26(2):195–219, 2000.

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

[57] A. Mulloni and V. Pekar. Automatic detection of orthographic cues for cognate recognition. *Proceedings of LREC'06, 2387*, 2390, 2006.

[58] L. Nakhleh, D. Ringe, and T. Warnow. Perfect phylogenetic networks: A new methodology for reconstructing the evolutionary history of natural languages. *Language*, 81(2):382–420, 2005.

[59] L. Nakhleh, T. Warnow, D. Ringe, and S.N. Evans. A comparison of phylogenetic reconstruction methods on an Indo-European dataset. *Transactions of the Philological Society*, 103(2):171–192, 2005.

[60] F.J. Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on ACL-Volume 1*, pages 160–167. ACL, Morristown, NJ, USA, 2003.

[61] F.J. Och and H. Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, 2003.

[62] K. Papineni, S. Roukos, T. Ward, and WJ Zhu. BLEU: a method for automatic evaluation of MT. *Research Report, Computer Science RC22176 (W0109-022), IBM Research Division, TJ Watson Research Center*, 17, 2001.

[63] Taraka Rama, Anil Kumar Singh, and Sudheer Kolachina. Modeling letter to phoneme conversion as a phrase based statistical machine translation problem with minimum error rate training. In *The NAACL Student Research Workshop*, Boulder, Colorado, 2009.

[64] D. Ringe, T. Warnow, and S. Evans. Polymorphic characters in Indo-European languages. *Languages and Genes, September*, 2006.

[65] D. Ringe, T. Warnow, and A. Taylor. Indo-European and computational cladistics. *Transactions of the Philological Society*, 100(1):59–129, 2002.

[66] ES Ristad, PN Yianilos, M.T. Inc, and NJ Princeton. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532, 1998.

[67] R.J. Ryder. Grammar and Phylogenies.

[68] N. Saitou. The neighbor-joining method: a new method for reconstructing phylogenetic trees, 1987.

[69] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees, 1987.

[70] J. Schroeter, A. Conkie, A. Syrdal, M. Beutnagel, M. Jilka, V. Strom, Y.J. Kim, H.G. Kang, and D. Kapilow. A Perspective on the Next Challenges for TTS Research. In *IEEE 2002 Workshop on Speech Synthesis*, 2002.

[71] T. Sherif and G. Kondrak. Substring-based transliteration. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 944, 2007.

[72] Tarek Sherif and Grzegorz Kondrak. Substring-based transliteration. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 944–951, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[73] M. Simard, G. Foster, and P. Isabelle. Using Cognates to Align Sentences in Parallel Corpora. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 67–81, 1992.

[74] A. K. Singh and H. Surana. Can corpus based measures be used for comparative study of languages. In *Proceedings of the ACL Workshop Computing and Historical Phonology*, Prague, Czech Republic, 2007.

[75] Anil Kumar Singh. A computational phonetic model for indian language scripts. In *Proceedings of the Constraints on Spelling Changes: Fifth International Workshop on Writing Systems*, Nijmegen, The Netherlands, 2006.

[76] Anil Kumar Singh. Study of some distance measures for language and encoding identification. In *Proceeding of ACL 2006 Workshop on Linguistic Distances*, Sydney, Australia, 2006.

[77] A. Stolcke. Srilm – an extensible language modeling toolkit, 2002.

[78] H. Surana and A.K. Singh. A more discerning and adaptable multilingual transliteration mechanism for indian languages. In *Proceedings of the Third International Joint Conference on Natural Language Processing*, 2008.

[79] M. Swadesh. Lexico-statistic dating of prehistoric ethnic contacts: with special reference to North American Indians and Eskimos. *Proceedings of the American philosophical society*, pages 452–463, 1952.

[80] D.L. Swofford, G.J. Olsen, P.J. Waddell, and D.M. Hillis. Phylogenetic inference. *Molecular systematics*, 2:407–514, 1996.

[81] P. Taylor. Hidden Markov Models for Grapheme to Phoneme Conversion. In *Ninth European Conference on Speech Communication and Technology*. ISCA, 2005.

[82] K. Toutanova and R.C. Moore. Pronunciation modeling for improved spelling correction. In *Proceedings of the 40th annual meeting of ACL*, pages 144–151, 2002.

[83] A. van den Bosch and S. Canisius. Improved morpho-phonological sequence processing with constraint satisfaction inference. In *Proceedings of the Eighth Meeting of the ACL-SIGPHON at HLT-NAACL*, pages 41–49, 2006.

[84] A. van den Bosch and W. Daelemans. Do not forget: Full memory in memory-based learning of word pronunciation. *proceedings of NeMLap3/CoNLL98*, pages 195–204, 1998.

[85] R. Zens and H. Ney. Improvements in phrase-based statistical machine translation. In *HLT Conf. / NAACL*, pages 257–264, Boston, MA, May 2004.