8.2 Study of Cognates among South Asian Languages for the Purpose of Building Lexical Resources

Anil Kumar Singh and Harshit Surana

Abstract— **High level of linguistic diversity in South Asia poses the challenge of building lexical resources across these languages. The only way we can hope to do this is by automating as much of this task as possible. This, in addition to the algorithmic aspect, also has a linguistic aspect in the sense that linguistic study can tell us what and how much can be automated. In this paper, we present a study of cognates across some South Asian languages for estimating how much of the task of building lexical resources can be automated. For identifying the cognates, we have used a unified computational model of scripts (UCMS) for Brahmi origin scripts. We have previously applied UCMS to solve several other practical problems. Based on the results of cognate identification, we suggest some implications for building lexical resources.**

Index Terms—**Cognates, South Asian languages, Building lexical resources, Unified Computational Model of Scripts.**

I. INTRODUCTION

CREATING large scale lexical resources is as difficult in terms of time and effort required, as it is important for building Natural Language Processing (NLP) applications or for linguistic reference. Given the number of major languages in India and the lack of financial and other resources, it may not be very practical to build lexical resources manually in the conventional ways. We need innovative ways to create such resources which can make use of computational power. However, as we discuss in the next section, automatically creating resources like bilingual or multilingual dictionaries has not been very successful so far. This is because the task is quite difficult. To make this task easier, we need some linguistic insights. This is especially important in the South Asian context because South Asian languages have a lot of similarities [7] which can be abstracted out and used for computational purposes in solving problems which are otherwise not easy to solve.

In this paper we present a study of our experiments on automatic identification of cognates across some South Asian languages. We also suggest some implications of the result we obtained for building lexical resources.

The method used for identification of cognates is based on a Unified Computational Model of Scripts [17] that we have previously used for solving several practical problems like spell checking, text normalization, improving information retrieval, shallow morphological analysis [18] etc.

Anil Kumar Singh (email: anil@research.iiit.net) and Harshit Surana (email: surana.h@gmail.com) are with Language Technologies Research Centre, International Institute of Information Technology, Hyderabad, India

II. SOME RELATED WORK

There has been some work on writing systems [25] from the computational point of view. Sproat [21] presented a computational theory of writing systems. He also studied Brahmi scripts [19] and even performed a formal computational analysis of Brahmi scripts [20].

Some other related work is on phonetic modelling of graphemes. Rey et al. [12] argue that graphemes are perceptual reading units and can be considered the minimal 'functional bridges' in the mapping between orthography and phonology. Black et al. [1] have discussed some issues in building general letter to sound rules within the context of speech processing.

Emeneau [7], in his classic paper 'India as a Linguistic Area', showed that there are a lot of similarities among Indian languages, even though they belong to different families.

Our model of alphabet, which is a part of the UCMS, is based on the traditional knowledge about the scripts used for Indian languages and the work done on encodings for Indian languages. Perhaps the most important work in this category is the development of a standard for Brahmi origin scripts [4, 5], called Indian Standard Code for Information Interchange (ISCII). This has also been called a super-encoding or meta-encoding. It took into account the similarities among the alphabets of Brahmi origin scripts.

Singh [16] had proposed a computational phonetic model of Brahmi based scripts based on orthographic and phonetic features. These features were defined based on the characteristics of the scripts. The similarity between two letters was calculated using an SDF and the algorithm used for 'aligning' two strings was dynamic time warping or DTW [11].

The unified model [17] also takes into account non-phonetic aspects of Brahmi scripts, like the aaksharik nature of these scripts and uses a very different way for calculating surface similarity.

The need for automatically extracting dictionaries has been recognized for a long time, which is natural since building dictionaries for various language pairs (especially in electronic machine readable form) is a long and difficult task for humans. There have been many attempts in this direction but the accuracies achieved so far have not been very high.

Perhaps the biggest systematic effort at building multilingual dictionaries is the Papillon project [2]. It aims at 'creating a cooperative, free, permanent, web

oriented environment for the development and the consultation of a multilingual lexical database'. But rather than separately creating dictionaries for different language pairs, it uses a set of monolingual dictionaries of word senses (lexies) linked through a central set of interlingual links (axies).

This linking of monolingual dictionaries can be done automatically [22]. Automatic extraction can also be done from comparable corpora [14]. Attempts have also been made to create `WordNet-like lexical databases' [9].

An early attempt was Daelemans's [6] tool for automatic creation, extension and updating of lexical knowledge bases distinguishes between two levels of representation: a static storage level and a dynamic knowledge level.

Among others, Schiffman and McKeown [15] tried automatically building a lexicon of phrases from a collection of documents for question answering.

Verma and Bhattacharyya [23, 24] have tried to automatic generate multilingual lexicon by using WordNet. Baud et al. [3] describe a method to facilitate the interchange of lexical information for multiple languages in the medical domain. Farwell et al. [8] have used a method for automatic creation of lexical entries for a multilingual machine translation system.

Ribeiro et al. [13] have surveyed some of the algorithms for cognate alignment, including that by Melamed [10]. Their method is based on finding identical words as well as typical contiguous and non-contiguous character sequences extracted using a statistically sound method. Since they used this method for alignment of parallel text, we have not used it for comparison with our method as we are using non-parallel corpus.

III. SOUTH ASIA AS A LINGUISTIC AREA

South Asia has a common historical and cultural background. This is also reflected in the languages of this area. That most of these languages, in spite of belonging to various families, have a lot of similarity, was established formally in Emeneau's classic work [7]. From this work has emerged the idea of 'India as a linguistic area'. This occurrence is also called the South Asian convergence. So, in fact, it is better to talk about South Asia as a linguistic area.

There has been a debate about the reasons for this convergence, but whatever they might be, similarity among the South Asian languages is an established fact. This is understandable given the long term contact, migrations of populations, common historical and cultural background etc. It is as if South Asian languages form a family of their own, which cuts across different conventionally identified linguistic families.

IV. COGNATES AMONG SOUTH ASIAN LANGUAGES

Due to the 'convergence', South Asian languages have a lot of cognate words, i.e., their vocabularies have a significant overlap. For example, a lot of words in these languages have been borrowed from Sanskrit. Some of them retain their original form (tatsam), while others have changed their forms (tadbhav). Similarly, some of them have retained their meanings, while others have become associated with different concepts.

Apart from the large number of Sanskrit words, there are also a lot of words borrowed from languages which have been or still are dominant at some time in the history of South Asia. Persian and English are two such languages. Other 'foreign' languages like Arabic, Turkish and Portuguese etc. have also contributed their words to the South Asian languages, though in smaller numbers. Then there are the words which South Asian languages have borrowed from each other or from extinct or nearly extinct tribal and minority languages or dialects. Still another category of words which may be considered cognates are the onomatopoeic words.

Identifying such cognate words can be a major step in calculating cross lingual lexical similarity in general. And, of course, taking care of cognates can help us in reducing the work involved in building lexical resources.

V. SOME EXAMPLES

We can roughly divide the cognates into two categories. In the first category are those cognates which have more or less the same meaning in the two languages, while in the second the meanings have changed.

Some examples of cognates extracted from our method are given below.

- **Same origin, same meaning :**
  - Hindi-Bengali : हाजार-হাজার, जीवन-জীবন, दोकानदार-দুকানদার, ओलट-উলটা, सोहाग-সুহাগ, कालेक्टर-কলেক্টর etc.
- **Same origin, different meaning :**
  - Hindi-Bengali : One very good example of this is अभिमान This word is used in many Indian languages but has different meanings in different languages.

These two categories of cognates have to be handled differently for practical computational applications. While the first can be directly used for building lexical resources, the second cannot be used directly.

VI. LEXICAL SIMILARITY AT THE SURFACE

Surface similarity can be divided into two overlapping parts: orthographic and phonetic. For calculating such similarity, we have used a unified model of scripts [17]. This model is very useful for South Asian languages because of the phonetic and aaksharik (loosely, syllabic) nature of Brahmi origin scripts.

Earlier work on connecting phonology and orthography has focused on letter to phoneme (or vice-versa) mapping. We have used a model of scripts based on phonetic and orthographic features of letters, a stepped distance function (SDF), the aaksharik nature of the scripts and a shallow model of morphology, etc. We can use either the Computational Phonetic Model of Scripts (see that next section) alone or we can use the UCMS. The phonetic and orthographic features are manually defined based on the characteristics of the scripts. In the CPMS, similarity of two strings can be calculated by using a dynamic programming algorithm called the dynamic time warping (DTW) algorithm.

VII. UNIFIED COMPUTATIONAL MODEL OF SCRIPTS

The UCMS [17, 18] aims to capture the characteristics and commonalities of a group of scripts, restricted to the Brahmi origin scripts for the time being. The idea is that instead of only calculating literal string similarities or even phonetic similarities, we can use all the information that would become available to the computer if the computer knew about the linguistic characteristics of the scripts. This would make out applications more accurate and more flexible, and these applications might work well across all the languages which use the scripts covered by the model.

The schematic diagram of the UCMS is shown in figure-1.

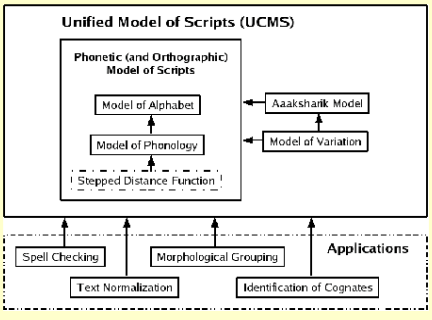

Fig.-1: Unified Computational Model of Scripts

As can be seen from the figure, the unified model consists of various component models. The most important part of the UCMS (for Brahmi scripts) is the Computational Phonetic Model of Scripts (CPMS). The CPMS is itself composed of a model of alphabet and a model of phonology, a stepped distance function for calculating the similarity of letters or akshars, an alignment algorithm for calculating the similarity of words or strings. Other parts of the UCMS are a model of variation and a shallow model of morphology (Singh and Surana, 2007a). The latter has not been shown in the figure. Some of the applications for which we have already used the UCMS are shown in the figure.

VIII. AUTOMATICALLY IDENTIFYING COGNATES

The cognate identification or extraction algorithm was based on using the UCMS. Surface similarity scores were calculated for pairs of words from different languages and a threshold was applied. The search was fast enough because we are using an FST of akshars. A dynamic programming based algorithm was used for aligning strings on the FST. Note that this algorithm is different from the one used in CPMS. The CPMS was used for calculating akshar pair similarity scores. Since these scores were used while aligning the strings, these calculations were performed only once. The list of possible akshars was extracted from the corpus.

For our experiments, we have used the ERDC parallel corpus. The cognate extractions algorithm was run on word lists (along with frequencies) extracted from the corpus. Only the top 20000 or so words were used for identifying cognates.

IX. COGNATE IDENTIFICATION EVALUATION

Since it was not possible to extensively prepare manual reference data across all the language pairs, we used random sampling to evaluate cognate extraction. We extracted cognates by applying a threshold and then randomly extracted 200 candidate cognates from them. These lists of 200 cognates were manually checked by people who knew the two relevant languages quite well.

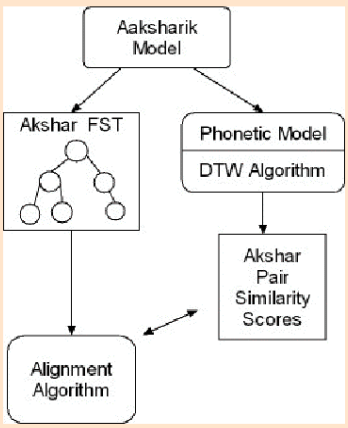

Figure 1: Identifying Cognates Using the UCMS

X. ESTIMATES OF COGNATE COVERAGE

By coverage of cognate coverage we mean what percentage of words in documents in South Asian languages are cognates for particular pairs of languages. Table-1 gives rough estimates of the coverage of cognate among several language pairs. Right now we have not calculated the coverage of cognates across more than two languages.

The estimates are only rough estimates because we have extrapolated from the results of our evaluation of the cognate identification algorithm and these estimates might include proper nouns (around 5% of extracted cognates). The validity of these estimates may also be different for different language pair because the algorithms used for cognate identification are, naturally, asymmetric. However, to counter this problem, we tried to make the algorithm as symmetric as possible. This means that no language-pair specific tuning of the algorithm was performed for the purpose of calculating cognate coverage, even though we could have used such tuning to improve the accuracy of identification for certain pairs of languages.

Since the size of the ERDC corpus for different languages was different, we compensated by performing the calculations only for a particular range of frequencies. The token list size for each language was more than 20000 words.

XI. IMPLICATIONS FOR BUILDING LEXICAL RESOURCES

One inference of the results of cognate identification that is clear from table-1 is that the percentage of cognates that can be extracted automatically depends on the similarity (or distance) between the two languages. For example, Hindi and Punjabi or Hindi and Bengali are much closer than Hindi and Telugu or Telugu and Marathi. Accordingly, many more cognates can be extracted for the first two pairs than for the last two. This is along the expected lines. Still, the results give us a quantitative estimate of the coverage of cognates. However, it is important to note that these results are from a particular corpus and are valid more for written language. Spoken language is likely to have a different distribution of cognates, e.g., there will be fewer Sanskrit origin tatsam words in spoken language, but more words of English or Persian origin. Another important point is that these figures are for tokens, not types. Since we are mostly interested in written language, these results can still be useful for us.

Hindi-Punjabi	**57.63**
Hindi-Marathi	**44.54**
Hindi-Telugu	**28.86**
Hindi-Bengali	**50.43**
Telugu-Marathi	**28.62**
Telugu-Kannada	**34.67**
Bengali-Assamese	**55.48**

Table-1: Percentage Cognates (Tokens)

Another implication from our observation of the output is that some language pairs like Hindi-Telugu have cognates mostly in the category of tatsam words, whereas pairs like Hindi-Punjabi have cognates of other kinds too. This means that the task of preparing practically useful lexical resources is even more difficult than what appears from the figures given in table-1.

The results also show that if surface similarity is calculated in a more linguistically aware way, then just by calculating such similarity we can get a lot of crosslingual information that can be used for building lexical resources.

A surprising result is that Telugu and Kannada have significantly fewer cognates than Hindi-Marathi, though the first two are linguistically and geographically supposed to be closer than the last two. If our figures are valid, and not heavily biased by the corpus, then it implies that the ease of building lexical resources may not be in direct proportion to the linguistic distance between two languages.

For building multilingual lexicon, our results show that we have to rely on more intelligent applications for extracting corresponding words across languages. Such applications have to take into account the characteristics of languages or language pairs. An algorithm that has no way to incorporate linguistic knowledge will not work equally well for all languages or language pairs. The challenge is to design applications which can do this with minimum human intervention.

More specifically, as mentioned earlier, the two categories cognates have to be handled differently. Since the second kind of cognates, i.e. words which occur across different languages but with different meanings, we need to combine the linguistics knowledge based approaches with machine learning based approaches. We could distinguish between the two kinds of cognates and the meanings of the second kind by using a method based on contextual similarity, somewhat like the methods used for word sense disambiguation.

XII. CONCLUSION

In this paper we have presented a study of the results of automatic cognate extraction from non-parallel multilingual corpora of some South Asian

languages using a Unified Computational Model of Scripts, which we have previously applied to several other practical problems. We argue that these results have some implications for building lexical resources, both from linguistic and computational points of view. Taking care of these implications might make the task of building lexical resources easier, given the scarcity of financial and other resources for this task and the high linguistic diversity in the South Asian region.

REFERENCES

[1] Black, A.; Lenzo, K.; and Pagel, V. 1998. Issues in building general letter to sound rules. In ESCA Synthesis Workshop, Australia., 164171.

[2] Boitet, C., Mathieu Mangeot-Lerebours, M. and Gilles, S. 2002. The PAPILLON Project: Cooperatively Building a Multilingual Lexical Database to Derive Open Source dictionaries and Lexicons. Proc. of the 2nd Workshop NLPXML 2002, Post COLING 2002 Workshop. Taipei, Taiwan.

[3] Baud, R. H., Nystram, M., Borin, L. Evans, R., Schulz, S. and P Zweigenbaum. 2005. Interchanging Lexical Information for a Multilingual Dictionary.

[4] Bureau of Indian Standards. 1991. Indian standard code for information interchange (ISCII).

[5] C-DAC. 2006. Standards for Indian languages in it. http://www.cdac.in/html/gist/standard.asp.

[6] Daelemans, Walter. 1987. A tool for the automatic creation, extension and updating of lexical knowledge bases. Proceedings of the third conference on European chapter of the Association for Computational Linguistics. pp 70-74. Association for Computational Linguistics.

[7] Emeneau, M. B. 1956. India as a linguistic area. In Linguistics 32:3-16.

[8] Farwell, D., Guthrie, L. and Wilks, Y. The Automatic Creation of Lexical Entries for a Multilingual MT System. In Proc. of COLING 2002.

[9] Lipatov, A., Goncharuk, A., Helfenbein, I., Shilo V. and Lehelt, V. 20003. Automatic Creation of Non-English WordNet-like Lexical Databases. Papillon 2003 Workshop. NII, Tokyo, Japan.

[10] Melamed, I. D. 1997. A portable algorithm for mapping bitext correspondence. In Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, 305312. Association for Computational Linguistics.

[11] Myers, C. S., and Rabiner, L. R. 1981. A comparative study of several dynamic time-warping algorithms for connected word recognition. In The Bell System Technical Journal, 60(7), 13891409.

[12] Rey, A.; Zieglerc, J. C.; and Jacobse, A. M. 2000. Graphemes are perceptual reading units. In Cognition 74.

[13] Ribeiro, A.; Dias, G.; Lopes, G.; and Mexia, J. 2001. Cognates alignment. Machine Translation Summit VIII, Machine Translation in The Information Age 287292.

[14] Sadat, F. 2002. A Combination of Models for Bilingual Lexicon Extraction from Comparable Corpora. Papillon 2002 Seminar. NII, Tokyo, Japan.

[15] Schiffman, B. and McKeown, K. 2000. Experiments in automated lexicon building for text searching. In Proc. of COLING-2000.

[16] Singh, A. K. 2006. A computational phonetic model for indian language scripts. Constraints on Spelling Changes: Fifth International Workshop on Writing Systems Nijmegen, The Netherlands.

[17] Singh, A. K. and Surana, H. 2007a. There can be Depth in the Surface: A Unified Computational Model of Scripts and Its Applications. Under submission.

[18] Singh A.K. and Surana, H. 2007b. Using a Model of Scripts for Shallow Morphological Analysis Given an Unannotated Corpus. ADD-2 Morpho-Syntactic Analysis (2nd School of Asian ANLP for Linguistics Diversity and Language Resource Development) , Pathum Thani, Thailand.

[19] Sproat, R. 2002. Brahmi scripts. In Constraints on Spelling Changes: Fifth International Workshop on Writing Systems.

[20] Sproat, R. 2003. A formal computational analysis of indic scripts. In International Symposium on Indic Scripts: Past and Future.

[21] Sproat, R. 2004. A Computational Theory of Writing Systems.

[22] Teeraparbseree, A. 2003. Jeminie: A Flexible System for the Automatic Creation of Interlingual Databases. Papillon 2003 Workshop. NII, Tokyo, Japan.

[23] Verma, N. and Bhattacharyya, P. 2003. Automatic Generation of Multilingual Lexicon by Using WordNet. International Conference on Convergence of Knowledge, Culture, Language and Information Technology. Library of Alexandria, Egypt.

[24] Verma, N. and Bhattacharyya, P. 2004. Automatic Lexicon Generation through WordNet. Global WordNet Conference (GWC-2004), Czech Republic.

[25] Coulmas, Florian. Writing Systems: An Introduction to their Linguistic Analysis. Cambridge University Press. 2003.

This paper was presented at LRIL-2007: National Seminar on Creation of Lexical Resources for Indian Language Computing and Processing at C-DAC Mumbai (26th to 28th March 2007), jointly organized by the Commission for Scientific and Technical Terminology (CSTT), New Delhi, MHRD, Govt. of India and the Centre for Development of Advanced Computing (C-DAC), Mumbai, Department of Information Technology, MC&IT, Govt. of India.