Learning Agglutinative Morphology of Indian Languages with Linguistically Motivated Adaptor Grammars

Arun Kumar
Universitat Oberta
de Catalunya
akallararajappan@uoc.edu

Lluís Padró
Universitat Politècnica
de Catalunya
padro@cs.upc.edu

Antoni Oliver
Universitat Oberta
de Catalunya
aoliverg@uoc.edu

Abstract

In this paper an automatic morphology learning system for complex and agglutinative languages is presented. We process complex agglutinative morphology of Indian languages using Adaptor Grammars and linguistic rules of morphology. Adaptor Grammars are a compositional Bayesian framework for grammatical inference, where we define a morphological grammar for agglutinative languages and morphological boundaries are inferred from a corpora of plain text. Once it produces morphological segmentation, regular expressions for orthography rules are applied to achieve final segmentation. We test our algorithm in the case of three complex languages from the Dravidian family and evaluate the results comparing to other state of the art unsupervised morphology learning systems and show significant improvements in the results.

1 Introduction

Morphological processing is an important step for natural language processing systems, specially for morphology-rich or agglutinative languages. In morphological processing a word is segmented in corresponding morphemes that are required for later stages of language processing. Most of the morphological systems are hand-built, which is a time consuming and costly process –e.g. finite state methods based morphology learning (Beesley, 1998). For this reason, least resourced languages lack this important component which act as major hurdle for building NLP systems. Unsupervised learning of morphology is a solution for dealing with this problem. In the case of unsupervised morphology learning systems –refer to (Hammarström and Borin, 2011) for details–, morphology of languages is learned using a corpus of plain text using statistical measures. Unsupervised morphology learning systems produce state-of-the-art results for many languages, such as English and Finnish (Creutz and Lagus, 2005; Goldsmith, 2001). In the case of Dravidian languages, poor results are obtained because of lack of knowledge of orthographical and morphological complexities, such as *Sandhi*, a morpho-phonemic change that happens in boundaries of word or morpheme concatenation.

Our method is a combination of statistical and rule based methods. The orthography related issues are solved by using a set of orthographic rules in the form of finite state transducer, which is rule based and morphological segmentation is achieved using statistical model of morphology based on Adaptor Grammar.

Adaptor Grammars are Bayesian non parametric models that can be used to learn linguistic structures. They are non parametric version of Probabilistic Context Free Grammar (PCFG). It is designed for unsupervised structure learning and successfully used in various natural language processing applications, such as word segmentation. We use Adaptor Grammars to learn model of morphology and once the model produce output we use regular expressions created from morphological rules and orthography to refine the results. The major idea behind the model is as these languages are agglutinated, suffixes are stacked to together to create a large word sequence. It indicates as the length of the word increase more number of morphemes are present in the word. With this intuition we define a model of morphology. We test our system on three major languages from Dravidian family. We choose three highly agglutinative and inflected languages from the family for experimentation such as Tamil, Malayalam and Kannada.

The structure of the paper as follows: In Section 2.1, we briefly discuss morphological prop-

Proceedings of Recent Advances in Natural Language Processing, pages 307–312,
Hissar, Bulgaria, Sep 7–9 2015.

erties and orthography of main languages in Dravidian family that make unsupervised learning difficult. In Section 2.2, we give an informal definition of Adaptor Grammars and inference procedure. In the following Section 4, we give the details of Adaptor Grammar model on Dravidian languages. The details of experiments and data used for experiments are explained in section 5. This section also includes a comparison of results with state of the art systems. Last section concludes the paper with future research directions.

2 Background

2.1 Challenges Related to Dravidian Languages

Dravidian languages are highly agglutinative like Turkish and inflected like Finnish. The major ones of the family are Tamil, Telugu, Kannada and Malayalam. They are major languages of southern part of India having millions of native speakers. In this study we focus on Tamil, Kannada and Malayalam.

These languages use alpha syllabic writing systems in which vowels are represented in the form of dependent symbols to consonant symbols i.e. consonant symbols are ligatures. The vowel symbols also can occur in the atomic form if they are not connected to consonants. For example, in the case of Malayalam (Taylor and Olson, 1995), (ക ka) represents a consonant ligature consisting of (ക് k) and a short vowel (അ a). As these languages use alpha syllabic writing system, symbols are syllables instead of individual characters like in English, and phonological changes occur during the concatenation of morphemes and words, resulting in a change in orthography. This process refereed as *Sandhi changes*. For example (Steever, 1998), in the Tamil word (நரி, nari, *fox*) + (ஆ ā) → (நரியா nariyā, *Is it a fox?*). The syllable (ஆ ā, *(interrogative)*) suffers a change in script and is changed to (யா, Yā).

They also contain large number of diacritics and digraphs. As a result, morpheme boundaries are marked at syllabic level. Dravidian languages are agglutinative and generate long word sequences with orthographic changes. All these languages are highly inflected: nouns inflected with cases, gender, number and post positions. Verbs are inflected with tenses, mood , aspect and gender.

- (எழுத்துப்பூர்வமாக, Eḻuttuppūrvamāka, *writing*)
- (പൂർത്തിയാക്കിക്കഴിഞ്ഞു , pūrttiyākkikkaḻiññu, *finished*)

Compounding is another challenge in unsupervised morphology learning of these languages. Dravidian languages can generate a large number of compounds, which can in turn become components of larger compounds (Mohanan, 1986). For example; a compound word from Malayalam (ammāyiamma, *mother-in-law*) is a combination of two stems (ammāyi, *aunt*) and (amma, *mother*).

These languages contain co-compounds and sub-compounds with phonological changes (Inkelas, 2014), Some examples from Malayalam:

- kāṭṭilemaraṁ *forest-tree* (tree forest)
- tīvaṇṭi *fire-vehicle* (train)

Here the word kāṭṭilemaraṁ is a sub-compound since it is a combination of stems (kāṭṭile - *forest*, and maraṁ - *tree*). The word tīvaṇṭi is a co-compound because it contains just a head (vaṇṭi - *vehicle*) and a modifier (tī - *fire*). Named entities and proper names are also inflected and agglutinative with cases and number markers. It becomes more difficult when the named entity is a loaned foreign word. E.g., in Malayalam kampyūṭṭaṛinṛe is the combination of an English word *computer* and Malayalam genitive case marker.

2.2 Adaptor Grammar

An Adaptor Grammar is a 7-tuple $\langle N, W, R, S, \theta, A, C \rangle$, where $\langle N, W, R, S, \theta \rangle$ is a PCFG with a set of non terminals N, a set of terminals W, starting symbol $S \in N$, and a set of rewriting rules R where each $r \in R$ has probability $\theta_r \in \theta$. $A \subseteq N$ is a set of *adapted non terminals*, and C is a vector of adaptors indexed by elements of A, such that C_X is an adaptor for adapted non terminal $X \in A$, that is, C_X is a mapping from the set T_X of sub trees with root X to a base probability distribution H_X, determined by the probabilities of PCFG rules expanding X. Inference on the model is done using Markov Chain Monte Carlo techniques. For technical details see (Johnson et al., 2007)

Various non parametric probabilistic processes can be used as adaptors, like Dirichlet Process. Johnson uses Dirichlet Process as adaptor for word segmentation of Sesotho (Johnson, 2008).

3 Related Work

Recent research in morphology learning has shifted to semi-supervised learning, obtaining better results than fully unsupervised learning, such as (Kohonen et al., 2010a) and (Kohonen et al., 2010b). But In the case of Indian languages unsupervised morphology is rarely applied. As state of the art morphology learning systems give good results in the case of European languages, there are some efforts to test Dravidian languages on these systems. But obtained results are rather poor (Bhat, 2012). These studies give an idea of a rule and statistics based model that can work well on Indian languages. In the case of Adaptor Grammars, they are applied to various NLP tasks such as: Word segmentation (Johnson, 2008), named entity recognition (Elsner et al., 2009), and machine transliteration (Wong et al., 2012).

4 Adaptor Grammars on Dravidian Languages and Inference Procedure

We use a Pitman-Yor process based adaptor (Pitman, 2002) for learning the complex morphology of Dravidian languages. Pitman-Yor process is a stochastic model that can be represented in the form of a Chinese Restaurant Process metaphor. This representation helps to do inference on the model. The Pitman-Yor process is used as an adaptor that means our non terminals are placed with prior distribution, which is Pitman-Yor process.

We define a model similar to (Goldwater et al., 2005), but our model is more complex because they consider that one word is composed of one stem and one suffix, which is not a valid assumption in the case of agglutinative languages. In the case of agglutinative languages, many suffixes can be stacked together to form a word consisting of many morphemes. Considering this factor we define a complex model where a stem can be followed by many suffixes. For instance, an agglutinative word phrase from Malayalam sansthānaṅṅaḷileānnāṇ can be represented in a PCFG trees. A PCFG tree can represent any segmentation of a particular word phrase like san + sthāna + ṅṅaḷil + eānnāṇ, but we need only the right morpheme segmentation, in this case it is sansth + ānaṅṅaḷileā + nnāṇ. Adoptor Grammar enable us to learn these tree fragments and allows to define a general model of morphology. From this we define a general model of the morpheme structure of the languages. We model agglutinative morphology using the following grammar:

$$
\begin{aligned}
Word &\rightarrow Stem \mid Stem\ Suffixes \\
Stem &\rightarrow chars \\
Suffixes &\rightarrow Aspect \\
Suffixes &\rightarrow Tenses \\
Suffixes &\rightarrow Mood \\
Suffixes &\rightarrow Case \\
Suffixes &\rightarrow Gender \\
Suffixes &\rightarrow Gender\ Number \\
Suffixes &\rightarrow Gender\ Case \\
Suffixes &\rightarrow Number\ Case \\
Gender &\rightarrow chars \\
Number &\rightarrow chars \\
Case &\rightarrow chars \\
Aspect &\rightarrow chars \\
Tenses &\rightarrow chars
\end{aligned}
$$

where $chars$ is any sequence of characters, and $Suffixes$ is an adapted non terminal. We place a Pitman-Yor process prior on $Suffixes$ non terminals. So it can be expanded according to rewrite rules and a rule probability defined by the Pitman-Yor Process. $Case$, $Gender$, $Number$, $Tense$ and $Mood$ are adapted non terminals, which represent the morphological variations. Each $Case$, $Gender$, $Number$, $Tense$ $Aspect$ and $Mood$ act as a *submorph* as in (Sirts and Goldwater, 2013) and this grammar is similar to compounding grammar described by them. The tree based model can represent all possible segmentation. But we place a Pitman-Yor process on the adapted non terminal $Suffixes$. It enables the expansion of PCFG tree in two ways: one based on the PCFG rule probability, and another based on rule probability sampled from the Pitman-Yor adaptor. Because of the caching property of the Pitman-Yor process, frequent morphemes are clustered together in tables of the Chinese Restaurant process. It also important to note that word morphemes are not completely independent entities, since there are various interdependencies between them. We consider bi-gram dependencies so the grammar we defined above is similar to *Collocation Adaptor Grammar* described in (Johnson, 2008), where terminals are syllables. We use the Metropolis Hasting inference algorithm described in (Johnson and Goldwater, 2009) for the inference procedure.

4.1 Rule Based Transliteration

We use a simple program that handles orthography and *Sandhi* rules. The main idea of the program

	Tamil	Kannada	Malayalam
Token frequency	500K	500 K	500 K
No. Segmented tokens	10K	10K	10K
No. R E expressions	34	62	34
No orthographic rules	89	67	96

Table 1: Corpus information

is to transliterate between native syllabic script in original texts, and phonetic romanized transcript. The program works in both directions, and is used to interface the Adaptor Grammar model with the input/output texts.

For example: an agglutinated Malayalam word phrase അടയാളപ്പെടുത്തുകയായിരുന്നു is converted to corresponding ISO romanized form which is (aṭayāḷappeṭuttukayāyirunnu, *have been marked*) to get unique phonological representation. And the unique form is converted to its syllabic structure. It is possible to track the Sandhi changes by converting the script into syllabic form. The changes can be insertion, deletion or substitution of syllables (e.g. മഴ + ആണ് → മഴയാണ്, ma̲ḻayāṇ, *raining*). When we convert the ISO form of the corresponding word all the syllables that are inserted (യ *y*) and the vowel in atomic form (ആ *ā*). This distinction is important for handling Sandhi.

We created rules for these orthographic conversions. If a syllable indicating vowel is inside the segmentation we transliterate the corresponding syllable to a dependent vowel. Otherwise the vowel symbol in atomic form is produced. Rules handle also consonant digraphs: if two consonants are together with a marker of diacritic syllable we produce a digraph character instead of individual consonants. For example: when a word such as gujar̲āttt is encountered, we convert the syllable tt as ത്ത instead of individual t symbols (ത ത).

We apply this conversion rules for two purposes: at first for conversion of orthographic script to syllabic form to fed it to the adaptor sampler and after sampling converting the syllables back to corresponding orthographic scripts.

5 Data and Experiments

For testing our method, we have extracted from Wikipedia and news websites a corpus of five million words of each language, and normalized the fonts to Unicode 6.1 version. The overall corpora was converted to 8-bit extended ASCII transcription using rule based transliteration. The conversion script works as follows: A Malayalam word (e.g. തൂത്തുകൾ) is converted to tutarcl, where Unicode character ṭ is converted to ASCII character t, which represents a syllable in our internal representation. We keep a single space between syllables. These syllables act as terminals of our PCFG trees. We convert 500K unsegmented tokens of each language as described above. Named entities and proper names are not removed, as they can also be inflected. Then we ran Adaptor Grammar model and inference algorithms for 100 iterations, and the sampled syllables are fed to the transliteration module, which produces the corresponding orthographic form.

For the evaluation of presented algorithms, we have morphologically segmented 10K words of each language[1], which is manually created. The details of corpus used in table presented in the Table 1, The evaluation is based on how well the methods predict the morpheme boundaries and calculated as precision, recall and F-score. Information of data used for experiments is provided in Table 1. We used python suite provided in the morpho-challenge website for evaluation purposes. We also trained as baselines Morfessor[2], Morfessor-CAP[3], and Morepheme++[4] with the same amount of tokens. As these software perform very well for inflected and agglutinative languages, such as Finnish and Turkish. All the software except Morepheme++ trained with 500 K tokens and models are created. The trained models applied to our 10K words in unsegmented form and evaluated the results with their morphological segmentation in orthographic form. In the case of Morpheme++, we ran the software on the 10K test tokens and compared the results with its corresponding morphological segmentation in ortho-

[1] Available in http://anonymized-URL

[2] https://pypi.python.org/pypi/Morfessor

[3] http://www.cis.hut.fi/project/morpho/morfessorcatmap-downloadform.shtml

[4] http://www.hlt.utdallas.edu/~sajib/Morphology-Software-Distribution.html

graphical form as the system is not model based.

The result of the experiment is presented in Table 2. It includes precision (P), recall (R) and F-score (F) based on the morphological segmentation produced in orthographic levels. We have performed a manual analysis of results to understand the improvement in precision and the errors.

- Since our method uses a rule based transliteration module, it handles better the orthography, which is very important. Other systems do not considering the digraphs as single entities, and thus, they wrongly segment the digraphs, resulting in lower performance.
- Also, our system is the only that handles *Sandhi* changes.
- In the case of Kannada all systems show good performance because the language has a smaller amount of digraphs.
- When the word stem is a loaned word, all systems failed to segment it.

6 Conclusion and Future Research

We have presented a semi supervised morphology learning technique that uses Adapter Grammars and linguistic rules. And the result show that a method that is a combination of statistical and rule based method can give better performance than a fully unsupervised method in complex languages. We also show that handling orthography of Indian languages using rules is useful for handling morpho-phonemic complexities. In the future research, we will extend the system to other languages in the Dravidian family such as Tulu and Telugu.

References

Kenneth R Beesley. 1998. Arabic morphology using only finite-state operations. In *Proceedings of the Workshop on Computational Approaches to Semitic languages*, pages 50–57. Association for Computational Linguistics.

Suma Bhat. 2012. Morpheme segmentation for kannada standing on the shoulder of giants. In *24th International Conference on Computational Linguistics*, page 79.

Mathias Creutz and Krista Lagus. 2005. *Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0*. Helsinki University of Technology.

Micha Elsner, Eugene Charniak, and Mark Johnson. 2009. Structured generative models for unsupervised named-entity clustering. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 164–172. Association for Computational Linguistics.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational linguistics*, 27(2):153–198.

Sharon Goldwater, Mark Johnson, and Thomas L Griffiths. 2005. Interpolating between types and tokens by estimating power-law generators. In *Advances in neural information processing systems*, pages 459–466.

Harald Hammarström and Lars Borin. 2011. Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350.

Sharon Inkelas. 2014. *The interplay of morphology and phonology*. Oxford University Press.

Mark Johnson and Sharon Goldwater. 2009. Improving nonparametric Bayesian inference: Experiments on unsupervised word segmentation with Adaptor Grammars. In *naacl09*, pages 317–325.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *naacl07*.

Mark Johnson. 2008. Unsupervised word segmentation for sesotho using adaptor grammars. In *Proceedings of the Tenth Meeting of ACL Special Interest Group on Computational Morphology and Phonology*, pages 20–27. Association for Computational Linguistics.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010a. Semi-supervised learning of concatenative morphology. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 78–86. Association for Computational Linguistics.

Oskar Kohonen, Sami Virpioja, Laura Leppänen, and Krista Lagus. 2010b. Semi-supervised extensions to morfessor baseline. In *Proceedings of the Morpho Challenge 2010 Workshop*, pages 30–34.

Karuvannur P Mohanan. 1986. The theory of lexical phonology: Studies in natural language and linguistic theory. *Dordrecht: D. Reidel*.

Jim Pitman. 2002. Combinatorial stochastic processes. Technical report, Technical Report 621, Dept. Statistics, UC Berkeley, 2002. Lecture notes for St. Flour course.

Kairit Sirts and Sharon Goldwater. 2013. Minimally-supervised morphological segmentation using adaptor grammars. *Transactions of the Association for Computational Linguistics*, 1:255–266.

Method	Kannada			Malayalam			Tamil		
	P	R	F	P	R	F	P	R	F
Morfessor-base	67.92	59.02	45.63	38.21	41.59	48.54	41.75	41.59	41.78
Morfessor-CAP	70.32	53.77	62.1	62.64	47.11	53.77	59.48	41.07	44.09
Adaptor Grammar & rule	**73.63**	**59.82**	**66.01**	**65.66**	**54.32**	**59.66**	**53. 10**	**53. 17**	**53.13**
Morepheme ++	40.98	47.17	43.86	64.08	27.12	38.22	30.34	29.88	30.11

Table 2: Results; Compared to state of art systems

Sanford B Steever. 1998. *The Dravidian Languages*. Routledge London.

Insup Taylor and David R Olson. 1995. *Scripts and literacy: Reading and learning to read alphabets, syllabaries, and characters*, volume 7. Springer Science & Business Media.

Sze-Meng Jojo Wong, Mark Dras, and Mark Johnson. 2012. Exploring adaptor grammars for native language identification. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 699–709. Association for Computational Linguistics.