Sentence Boundary Detection in Kannada Language

Deepamala. N
Assistant Professor
Dept. of Computer Science
R.V. College of Engineering
Bangalore, India

Ramakanth Kumar. P
Professor and Head
Dept. of Information Science
R.V. College of Engineering
Bangalore, India

ABSTRACT
Sentence Boundary Detection is a pre-processing step for any Natural Language Processing application. Various algorithms have been used to achieve Sentence Boundary Detection or Disambiguation in different languages. In this paper, a rule based method is proposed and tested to achieve Sentence Boundary Detection for Kannada Language. Kannada being a grammatically rich Indian language is analyzed based on semantics and tested with a 227K bytes corpus. The code is written in C using wide characters, with support for Unicode. Results showed 99.2% success in detecting sentence boundary.

General Terms
Natural Language Processing, Kannada Language

Keywords
Sentence Boundary Detection, Verb Suffix, Abbreviation.

1. INTRODUCTION
Sentence Boundary detection is a preliminary step in any Natural Language Processing application. Many methods have been implemented and tested for Sentence Boundary Detection in English. Indian Languages being different in semantics from English Language requires different kind of approach. Kannada Language is a southern Dravidian Indian Language which is grammatically different from English. It is one of the 30 most spoken languages in the world. In this paper, a rule-based algorithm is proposed with results to detect Sentence Boundaries in Kannada Language Sentences. Sentence ending verb suffixes and Abbreviations are used as a parameter to classify the sentences.

2. LITERATURE REVIEW

2.1 Sentence Boundary Detection in English and other languages
Researchers have tried many algorithms and techniques to detect sentence boundaries of English Language. Many methods have been developed for Sentence Boundary Detection such as a rule-based sentence boundary detection algorithm by Manning et al. [1], using Maximum Entropy method by Reynar and Ratnaparkhi [2], Satz system by Palmer and Hearst [3] , and using POS tagging information by Mickheev [4].

Further, Kiss and Strunk propose a language independent method [5] by identifying abbreviations called Punkt Sentence Tokenizer. Walker et al.[6] compare three approaches for boundary detection. Yuya et al.[7] follow Statistical Language model (SLM) and support vector machine (SVM) approach to find sentence boundaries in Japanese language. Pritam et al. [8] propose algorithm using Maximum entropy and stop word algorithm. Many approaches have been followed to disambiguate sentence boundary in different languages by different researchers.

2.2 Sentence Boundary Detection in Indian Languages
Mona et al. [9] has developed a methodology to disambiguate period in Kannada Language by using lists of words below some threshold extracted from corpus. Very limited research has been undertaken in the area of Sentence Boundary Disambiguation for Indian Languages.

3. PRESENT WORK

3.1 Algorithm
The algorithm used for Sentence Boundary Detection in this paper is based on steps mentioned in [1] by Manning et .al. It is a heuristic Sentence Division algorithm which has the following steps:

- Place putative sentence boundaries after all occurrences of . ? ! (and maybe ; : -_)
- Move the boundary after following quotation marks, if any.
- Disqualify a period boundary in the following circumstances:
  - If it is preceded by a known abbreviation of a sort that does not normally occur word finally, but is commonly followed by a capitalized proper name, such as Prof. or vs.
  - If it is preceded by a known abbreviation and not followed by an uppercase word. This will deal correctly with most usages of abbreviations like *etc.* or Jr. which can occur sentence medially or finally.
- Disqualify a boundary with a ? or ! if:
  - It is followed by a lowercase letter (or a known name).
- Regard other putative sentence boundaries as sentence boundaries

In Kannada language, there is no concept of upper case or lower case letters. Hence, the above algorithm is modified and steps followed are listed below:

Step1: Place putative sentence boundaries after all occurrences of . ? ! ; : -_ . Let this be **Sentence1.**
- If Sentence1 is ? ! ; : - _ , regard the putative sentence boundary as sentence boundary.

Step2: Move the putative boundary after following quotation marks, if any, to next occurance of .?!;:- Let this be **Sentence2.**

Step3: Consider the last word of Sentence1 before period. Disqualify a period boundary of Sentence1 in the following circumstances:

- If period is preceded by a known abbreviation of a sort that does not normally occur word finally. Such abbreviations are listed in ABBREVIATIONS file.

Step3: Regard the putative sentence boundary of Sentence1 as sentence boundary

- If it matches with any of the verb forms that can possibly end a sentence, such verb suffixes are listed in VERBS_SUFFIX file.

Step4: Make Sentence2 as Sentence1 and Repeat from Step2.

The Sentence Boundary Detection algorithm proposed in this paper uses 2 files which are as discussed below:

ABBREVIATIONS File:

The Abbreviations file contains a list of abbreviations listed from Kannada newspapers like ಪ್ರೊ. (Prof.), ಡಾ. (Dr.) and Kannada translation of English alphabets like ಎ., ಬಿ., ಸಿ. … (A,B,C….). They are used as Initials before the First Name of a person. E.g.: ಎನ್. ದೀಪಮಾಲ (N. Deepamala).

VERBS_SUFFIX File:

All the 3 parts of speech take different form to indicate different tense or ಕಾಲ. The Sentence ending verb takes different forms based on its suffix. The VERBS_SUFFIX file contains the suffix form of different verbs. A Kannada sentence is divided into ಕರ್ತೃ ಪದ (Noun), ಕರ್ಮ ಪದ(Object). ಕ್ರಿಯಾ ಪದ(verb).

Eg: ರಾಮನು(Noun) ಕಾಡಿಗೆ(Object) ಹೋದನು(Verb).

Translation: Rama went to the forest.

The word preceding the period of a putative sentence is first verified with ABBREVIATIONS file and if it does not match, then with the list of suffixes in VERBS_SUFFIX file.

3.2 Verb Suffixes

The verb suffix forms based on tense are listed in Table 1. In Table 1, MG is masculine gender, FG is feminine gender and NG is neutral gender. Verb types and its suffixes based on meaning are listed in Table2. Some special verb suffixes are used to describe the task like When? How? As listed in Table 3 below:

Table 1. Verb classification with suffixes based on tense

Tense - ಕಾಲ	**Forms singular/ plural ವಚನ**	**Verb Stem + Tense Phrase + Verb Suffix = Inflected Verb ಧಾತು+ಕಾಲಸೂಚಕಪ್ರತ್ಯಯ+ ಅಖ್ಯಾತಪ್ರತ್ಯಯ = ಕ್ರಿಯಾಪದ**
Past Tense ಭೂತ ಕಾಲ	Singular – ಏಕವಚನ	ಹರಿ+ದ+ಎನು=ಹರಿದೆನು ಹರಿ+ದ+ಎ=ಹರಿದೆ ಹರಿ+ದ+ಅನು=ಹರಿದನು(MG) ಹರಿ+ದ+ಅಳು=ಹರಿದಳು (FG) ಹರಿ+ದ+ಇತು=ಹರಿಯಿತು (NG)
Past Tense ಭೂತ ಕಾಲ	Plural – ಬಹುವಚನ	ಹರಿ+ದ+ಎವು=ಹರಿದೆವು ಹರಿ+ದ+ಇರಿ=ಹರಿದಿರಿ ಹರಿ+ದ+ಅರು=ಹರಿದರು(MG) ಹರಿ+ದ+ಉವು=ತಿಳಿದುವು (NG)
Present tense ವರ್ತಮಾನ ಕಾಲ	Singular – ಏಕವಚನ	ಕೊಡು+ಉತ್ತ+ಎನೆ=ಕೊಡುತ್ತೇನೆ ಕೊಡು+ಉತ್ತ+ಈಯೆ=ಕೊಡುತ್ತೀಯೆ ಕೊಡು+ಉತ್ತ+ಆನೆ=ಕೊಡುತ್ತಾನೆ(MG) ಕೊಡು+ಉತ್ತ+ಆಳೆ=ಕೊಡುತ್ತಾಳೆ(FG) ಕೊಡು+ಉತ್ತ+ಅದೆ=ಕೊಡುತ್ತದೆ(NG)
Present tense ವರ್ತಮಾನ ಕಾಲ	Plural – ಬಹುವಚನ	ಕೊಡು+ಉತ್ತ+ಎವೆ=ಕೊಡುತ್ತೇವೆ ಕೊಡು+ಉತ್ತ+ಈರಿ=ಕೊಡುತ್ತೀರಿ ಕೊಡು+ಉತ್ತ+ಆರೆ=ಕೊಡುತ್ತಾರೆ(MG) ಕೊಡು+ಉತ್ತ+ಅವೆ=ಕೊಡುತ್ತವೆ(NG)
Future tense ಭವಿಷ್ಯತ್ಕಾಲ	Singular – ಏಕವಚನ	ಕೊಡು+ಉವ+ಎನು=ಕೊಡುವೆನು ಕೊಡು+ಉವ+ಎ=ಕೊಡುವೆ ಕೊಡು+ಉವ+ಅನು=ಕೊಡುವನು(MG) ಕೊಡು+ಉವ+ಅಳು=ಕೊಡುವಳು(FG) ಕೊಡು+ಉವ+ಉದು=ಕೊಡುವುದು(NG)
Future tense ಭವಿಷ್ಯತ್ಕಾಲ	Plural – ಬಹುವಚನ	ಕೊಡು+ಉವು+ಎವು=ಕೊಡುವೆವು ಕೊಡು+ಉವು+ಇರಿ=ಕೊಡುವಿರಿ ಕೊಡು+ಉವು+ಅರು=ಕೊಡುವರು(MG) ಕೊಡು+ಉವು+ಉವು=ಕೊಡುವುವು(NG)

Table 2. Verb classification with suffixes based on meaning

Verb Type	**Verb suffix**		**Example**
ವಿಧ್ಯರ್ಥ ಕ್ರಿಯಾಪದ	ಎನು ಉದು ಅಲಿ ಉ ಗೆ ಇ	ಇರಿ ಆಗು ಉವ ಓಣ ಆ	ಮಾಡುವೆನು ಮಾಡುವುದು ಮಾಡು ಮಾಡಲಿ ಮಾಡಿರಿ ಮಾಡೋಣ
ಸಂಭವವನಾರ್ಥಕ ಕ್ರಿಯಾಪದ	ಆನು ಆಳು ಈತು/ಆತು ಈಯೆ ಏನು	ಆರು ಆವು ಈರಿ ಏವು	ಬಂದಾರು ಬಂದಾನು ಹಚ್ಚೀನು ಹಚ್ಚೀವು ತಿಂದೀತು
ನಿಷೇಧಾರ್ಥಕ ಕ್ರಿಯಾಪದ	ಎನು ಎವು ಎ ಇರಿ ಅರಿ ಅನು	ಅರು ಅ ಅಳು ಅದು ಅವು	ಮಾಡೆನು ಮಾಡೆ ಮಾಡರು ಮಾಡಳು ಮಾಡದು

Table 3: Special verb descriptors

Verb description	Example
ಇದೆ	ಕಪ್ಪವಾಗಿ ಇದೆ
ಇವೆ	ಸಾಮಾನ್ಯವಾಗಿ ಇವೆ
ಇಲ್ಲ	ಮಾಡುವುದಿಲ್ಲ
ಅಲ್ಲ	ಮಾಡುವುದಲ್ಲ

3.3 Implementation

Sentence Boundary Detection algorithm for Kannada Language as discussed in the previous section is implemented using C language. Wide characters are used instead of characters to support Unicode. The C implementation of the software contains Wide character string operations like wcslen, wcstok, wcsspn, wcschr etc.

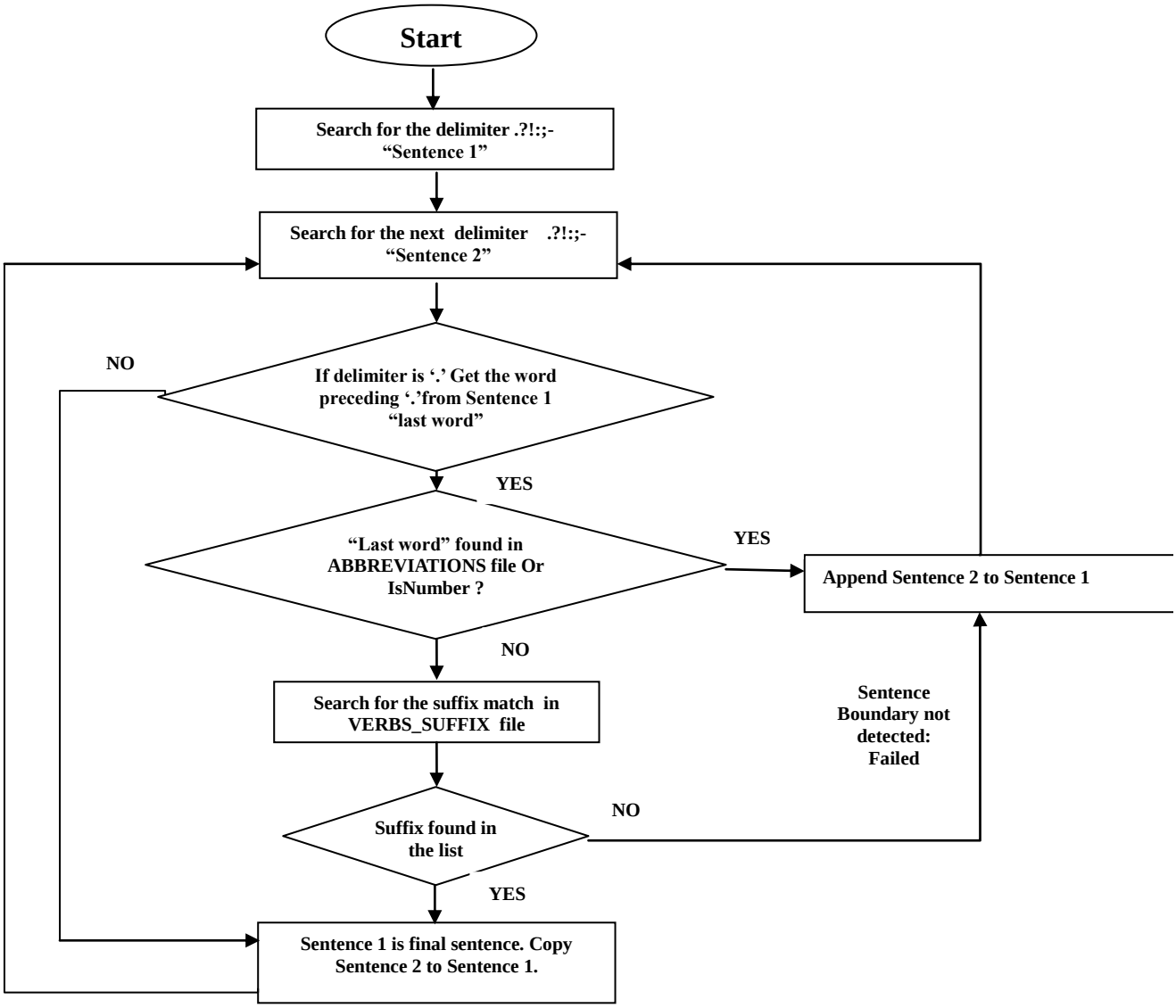

Fig 1: Sentence Boundary Detection Flow chart

4. DISCUSSION

There are many approaches followed to achieve sentence boundary detection. Very little research has been undertaken on sentence boundary detection for Indian Languages. Kannada, one of the Indian languages follows the grammar and syntax, which is completely different from English. For example, the rule that a period is followed by capital letter at the beginning of the sentence in English is used as a feature for sentence boundary identification [1, 5, 10, 11]. But, this feature is not applicable to Kannada language. [3, 4] use POS tagging information, but Sentence Boundary Detection can be a pre-processing task for a POS tagger. The algorithm proposed in Section 3 does not require POS tagging and rules are based on Kannada grammar.

Mona et al. [9] has explained disambiguation of sentence boundary for Kannada. Two lists, L1 and L2 are maintained, where L1 is a sentence ending word list and L2 is word list extracted from corpus. The comparison is made with L1 and L2 if last word length is below 5 (threshold). However, the author has not mentioned the Programming Language used for implementation.

The algorithm proposed in Section 3 is unique since a generic list of verb suffixes is maintained for comparison. Last word of any length in a sentence before period is matched with abbreviation file, and if not found, it is matched with the ending suffix. Substring match function is used to match the verb suffix with the ending word. If suffix matches, end of sentence is identified. The identified sentences are correct without ambiguity. Implementation using C wide characters makes the application more portable.

5. RESULTS

The developed application has been tested using EMILLE corpus. A corpus of 23,561 Kannada words (487KB) was given as Input to the Sentence Boundary Detection software, which detected 2152 sentences. Manually wrong sentences were identified and found that an accuracy of 99.2% is achieved with the software developed using the proposed algorithm.

The erroneous sentence boundary predictions were due to the following reasons:

- The '?' within a given sentence were wrongly predicted.
- If '.' Or '?' comes within quotes, they were wrongly predicted.
- If verb has no suffix, then the sentence is wrongly predicted.

6. CONCLUSION

Sentence Boundary Detection is a pre-processing step for any Natural Language Processing application. In the present implementation of Sentence Boundary Detection sentence ending verb and its different suffixes are used to detect the Boundary for Kannada Language. The result is almost 99.2% accurate. This technique is effective as no POS tagging or any other pre-requisite is required. As it is coded in C using wide characters, it is more portable. The same software can be used for similar Indian languages like Telugu by changing the ABBREVIATIONS and VERB_SUFFIX files accordingly.

7. REFERENCES

[1] Manning, C.D. and. Schütze., H. 2002. Foundations of statistical natural language processing. The MIT Press, London.

[2] J. Reynar, and Ratnaparkhi. A. 1997. A Maximum Entropy Approach to Identifying Sentence Boundaries, in Proceedings of the Fifth Conference on Applied Natural Language Processing, Washington D.C, pp. 16-19.

[3] Palmer, D.D. and Hearst, M.A..1997. Adaptive multilingual sentence boundary disambiguation. Computational Linguistics 23 241–267

[4] Mikheev, A. 2000. Tagging Sentence Boundaries. In: Proceedings of the NAACL, Seattle, pp 264-271.

[5] T. Kiss and Strunk, J. 2006. Unsupervised multilingual sentence boundary detection. Computational Linguistics, 32(4):485–525.

[6] Walker, Daniel J., David E. Clements, Maki, Darwin and Jan, W. Amtrup. 2001. Sentence boundary detection: a comparison of paradigms for improving MT quality. In: Proceedings of the MT Summit VIII, Santiago de Compostela, Spain.

[7] Akita, Y. 2006. Sentence Boundary Detection of Spontaneous Japanese Using Statistical Language Model and Support Vector Machines. In: Proceedings of. Interspeech-ICSLP, Pittsburgh, PA.

[8] Singh, Preetam, Negi, Rauthan M.M.S and Dhami, H.S. 2010. Sentence Boundary Disambiguation: a User Friendly Approach. IJCA. Vol, 7-No.8.

[9] Mona Parakh, Rajesha N. and Ramya M. 2011. Sentence Boundary Disambiguation in Kannada Texts, Language in India. www.languageinindia.com. 11:5 May 2011 Special Volume: Problems of Parsing in Indian Languages, pp. 17- 19.

[10] Gillick, D. 2009. Sentence Boundary Detection and the Problem with the U.S. In: Proceedings of the NAACL HLT: Short Papers, Boulder, Colorado.

[11] Agarwal N., Ford K., and Shneider M., Sentence Boundary Detection using a MaxEnt Classifier. citeseerx.ist.psu.edu

[12] Wang H. and Huang Y. 2003. Bondec - A sentence Boundary Detector. CS224N Project, Stanford, 2003