Sentence Boundary Disambiguation in Kannada Texts

Mona Parakh
Reader-Research Officer
ldc-monaparakh@ciil.stpmy.soft.net

Rajesha N.
Senior Technical Officer
ldc-rajesha@ciil.stpmy.soft.net

Ramya M.
Senior Technical Officer
ldc-ramya@ciil.stpmy.soft.net

Linguistic Data Consortium for Indian Languages
Central Institute of Indian Languages
Mysore, India
www.ldcil.org

Abstract - **The proposed paper reports the work on developing a system for identifying valid sentence boundaries in Kannada texts and fragmenting the text into sentences. The task of sentence boundary identification is made challenging by the fact that the period, question marks and exclamation marks, do not always mark the sentence boundary. This paper particularly addresses the issue of disambiguating period which can be a sentence boundary marker as well as a marker of abbreviation in Kannada. This methodology is devised to fragment corpora into sentences without any intermediate tools and resources like NER or Abbreviation List.**

I. Introduction

As an important and challenging task sentence boundary disambiguation (SBD) is the problem in natural language processing of deciding where sentences begin and end. Often natural language processing tools require their input to be divided into sentences for various purposes such as building bilingual parallel corpora. “A parallel corpus is a collection of texts in two languages, one of which is the translation equivalent of the other. Although parallel corpora are very useful resources for many natural languages processing applications such as building machine translation systems, multi-lingual dictionaries and word sense disambiguation, they are not yet available for many languages of the world” [2].

In order to process information from parallel text, it is first necessary to align the two texts at some level, typically at the level of paragraph or sentence. As in Reference [1], by ‘align’ is meant the association of chunks of text in the one document with their translation or equivalent text in the other document. In order to align text at the level of sentences, it is important to define and identify a sentence.

For the purpose of this work, we define a Sentence as a segment of text separated by delimiters such as Exclamation mark “!”, Question Mark “?”, Period “.” and new line character. However, these symbols do not always function as sentence delimiters; they can be used for other purposes, thereby making sentence boundary identification a non-trivial task. Sentence boundary identification is challenging because punctuation marks are often ambiguous.

Among the Indian languages Devanagari based scripts have the unique sentence boundary marker “।” known as ‘*poorna viraam*’ (full stop) which is different from the abbreviation marker - period. Hence, in such languages segmenting sentences is a relatively trivial task. But languages like English use period as a sentence boundary maker as well as abbreviation marker. As per the English examples given in Reference [2], “a period can also be used as a decimal point in numbers, in ellipses, in abbreviations and in email-addresses. The exclamation mark in the name of a web site Yahoo! may not signify a sentence boundary and so is the question mark in Which? - the name of a magazine”.

Like in English and many other languages even Kannada uses Period as a sentence boundary maker and for abbreviations. This paper attempts to handle this ambiguity of the Period in Kannada texts.

II. Method

Of the few papers that are available on work related to sentence boundary identification, Riley [4] uses a decision-tree based approach and claims a 99.8% performance on the Brown’s Corpus. Reynar and Ratnaparkhi [3] use a maximum entropy approach to identify sentence boundaries. Some of the other common algorithms for sentence boundary identification store the Standard abbreviation as a check list; however the approach proposed in this paper assumes that since abbreviations do not form a closed set, one cannot list all possible abbreviations.

In handling the ambiguity of period in this paper, we are considering the word length as a feature. Based on the study of Kannada corpus we can safely claim that it is usually the longer words that occur at the end of sentences. If a short word occurs with a period then it is most likely either an Abbreviation or a Salutation. Based on the corpus study, a minimal threshold for word length was decided. A list was created of words having length below the threshold and which were not abbreviations. A fairly exhaustive list of some 436 such words was obtained from (approx 4.2 million words) corpus. But the list was kept open-ended in order to accommodate further additions. However, after implementing the algorithm only a few Abbreviations which were above the threshold caused over segmentation of sentences.

The detection of abbreviations is an important step in the process of sentence boundary detection. Drawing upon Reference [5] abbreviations can be categorized into three classes TRAB, ITRAB and AMAB.

a) *TRAB*: These are transitive abbreviations, i.e., abbreviations that take an object and never end the sentence. To take an example from Kannada:

Kannada script:	ಮಿ. ಹರೀಶ್.
Transliteration:	mi. harIsh.
Translation:	Mr. Harish.

b) *ITRAB*: These are intransitive abbreviations that do not take an object. Even though Indian languages follow a relatively free word order in a sentence, normally Intransitive abbreviations do not come at the end of the sentence because, they are the subject of the sentence. Any intransitive abbreviation in the middle of a sentence will be handled by the algorithm. Following is an example from Kannada:

Kannada script: ತಮ್ಮ ಮೊಟ್ಟಮೊದಲಿನ ನಾಟಕವನ್ನು ಅ.ನ.ಕೃ. ೧೯೨೪ರಲ್ಲಿ ಬರೆದರು.

Transliteration: tamma moTTamodalina nATakavannu a.na.kx. 1924ralli baredaru.

Translation: A.Na.Kru. Wrote his first ever drama in 1924.

c) *AMAB*: These refer to abbreviations which are ambiguous, where a word is homonymous to an abbreviation.

Kannada script:	ಅದನಿಲ್ಲಿ ತಾ.
Transliteration:	adannilli tA.
Translation:	Bring that here.

Kannada script:	ತಾ. ೧೫-೦೮-೧೯೪೭
Transliteration:	tA. 15-08-1947
Translation:	Date. 15-08-1947

In the above example the verb ‘bring’ is homonymous to the standard abbreviation for ‘date’. “ತಾ.”/tA., could be the verb meaning “bring” occurring at the end of the sentence with a period marker or “ತಾ.”/tA., could be an abbreviation for “ತಾರೀಖು”/ tArIkhu meaning “date”.

III. Algorithm Design

Following is an algorithm devised to fragment the text into sentences by solving the ambiguity of period (“.”) as sentence marker and abbreviation in Kannada. The Algorithm uses two word lists as resource, viz. valid sentence ending word list (L1) and an ambiguous word list (L2) extracted from the corpus. This algorithm will disambiguate a period ending token as sentence ending word or abbreviation based on the token length. L1- has words having length below a threshold. L2- will have words with a length below a threshold and homonymous to an abbreviation of that language. Both L1 and L2 are extracted from corpus, and they make a small set of words. It should be noted that in this paper, the length of words refers to the length of Unicode characters and not the count of *aksharas*

Algorithm To Identify Period As Sentence Boundary

```
1. Preprocess the text in order to remove any space between a period (".") and its previous word.
2. Segment the text into sentences

1. Preprocess the text in order to remove any space between a period (".") and its   previous word.
          1.1 Open Text file
          1.2 Replace all "<space>." with "."

2. Segment the text into Sentences
 2.1 find the position of the Next Sentence Marker in the text
    2.1.1 WHILE starting position is less then Text length
    2.1.2 If the Next Immediate sentence Marker is "?" or "!" or New Line then Segment the text from Starting position to
          Sentence Marker
    2.1.3 If the Next Immediate sentence Marker is a period and not a Number before dot then
           2.1.3.1 Get the length of text between last space of text to period (Get the length of last word)
           2.1.3.2 If the Last word Length is below 5 (Threshold) then Check the word with L1
             2.1.3.2.1 If the Last word is in L1 then check the word with L2
                   2.1.3.2.1.1 If the Last word is not in L2 then Segment the text from Starting position to  Sentence Marker.
           2.1.3.3 If the Last word length is Equal or above threshold then check for the other possible dots in the Last word
              2.1.3.3.1 If there is no other possible dots in word then Segment the text from Starting position to Sentence
                             Marker.
              2.1.3.3.2 If there are other possible dots in word then check the Distance between   the end dot and the dot
                             end-but-one.
                     2.1.3.3.2.1 If Distance between the end dot and the dot end-but-one is above 5 (Threshold) then
                                      Segment the text from Starting position to Sentence Marker
    2.1.4 If the Next Immediate sentence Marker is a period and a Number before dot then Segment the text from Starting
                   position to Sentence Marker.
    2.1.5 End While

3. End
```

IV. Evaluation

In order to test the efficiency of the algorithm, a corpus of 7330 sentences (approx. 69000 words) was taken. Sentence Identification errors manually corrected and checked revealed that without using the algorithm and by a plain pattern matching of delimiters, a baseline accuracy of 91.33% was obtained. However, the accuracy increased to 99.14% after implementing the algorithm on the same corpus.

Out of the 7330 sentences in the corpus, the blind pattern matching without the algorithm showed errors in 636 sentences whereas after implementing the algorithm only 63 sentences were wrongly recognized. An increase of 7.81% from the baseline was noted after implementing the algorithm. The main errors occurred due to unclean corpus. Also, only a few Abbreviations which were above the threshold caused the over segmentation of certain sentences. The corpus used for the testing purpose was mainly from two domains – newspaper and literature.

V. Conclusion

In this paper we have described an algorithm for sentence boundary determination for Kannada. This methodology will hopefully be useful to resolve the problems of ambiguity of Period "." in case of text alignment tools, machine translation tools, KWIC KWOC Retrievers.

This method can be employed also for other languages. Since the check list used in the algorithm is open, it facilitates users to add more words to the list. However, depending on the language the length of the check lists may vary, as also the threshold.

Good performance has been obtained using this algorithm and it considerably increases the performance from the baseline.

Acknowledgment

Our special thanks to Prof. Kavi Narayana Murthy (CIS, UoH, Hyderabad; currently CIIL fellow) for his guidance, insightful comments and suggestions which helped us enormously in improving the evaluation work as also the paper. We are heartily thankful to our Project Head, Dr. L. Ramamoorthy, and the LDC-IL team members for their encouragement and support.

References

[1]. Harold Somers, "Bilingual parallel corpora and Language Engineering," in the Anglo-Indian Workshop on Language Engineering for South-Asian Languages, (LESAL), Mumbai, 2001.

[2]. Hla Hla Htay, G. Bharadwaja Kumar, and Kavi Narayana Murthy, "Constructing English-Myanmar Parallel Corpora," Proceedings of ICCA 2006: International Conference on Computer Applications, Yangon, Myanmar, pp 231-238, February 2006.

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

[4]. Riley, Michael D.. "Some applications of tree-based modeling to speech and language," in DARPA, Speech and Language Technology Workshop, Cape Cod, Massachusetts, 1989, pp. 339-352.

[5]. Trond Trosterud, Børre Gaup, Saara Huhmarniemi, "Preprocessor for Sámi language tools", The Norwegian Sami Parliament, 2004. Available online at www.divvun.no/doc/ling/preprocessor.html