19

Sentence Boundary Identification for Indian Language Texts: A Case Study of Kannada Texts

Mona Parakh
Rajesha N
Ramya M

Introduction: Sentence boundary identification also termed as sentence boundary disambiguation (SBD) is the problem in natural language processing of deciding where sentences begin and end. 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.

In languages which use Devanagari Script, period is not ambiguous, because the sentence boundary marker in Devanagari script is the “Devanagari Danda” [U+0964 known as poorna viraam (full stop)] whereas period is the marker for abbreviations. Hence, in such languages segmenting sentences is a relatively trivial task as compared to other languages that use period as a sentence boundary maker as well as for abbreviation marker. As per the English examples given in Htay et.al. (2006), “A period can also be used as a decimal point in numbers, in ellipses, in abbreviations and in email-addresses.”

“Sentence boundary disambiguation 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 number of reasons. However sentence boundary identification is challenging because punctuation marks are often ambiguous. For example, a period may denote an abbreviation, decimal point, an ellipsis, or an email address - not the end of a sentence.” (cf. http://en.wikipedia.org/wiki/Sentence_boundary_disambiguation)
The proposed methodology will be useful to resolve the ambiguity of the period in case of text alignment tools, machine translation tools, KWIC KWOC Retrievers.

For Western languages, this problem is dealt with in some approaches like checking the following word beginning with a capital letter (upper case character), storing the Standard abbreviation as a check list etc. But Indian Languages scripts do not have the distinction of upper or lower case characters, and as abbreviations do not form a closed set, one cannot list all possible abbreviations. Therefore, Indian languages need a different approach for disambiguating the period.

Method: Of the few papers that are available on work related to sentence boundary identification, Riley (1989) uses a decision-tree based approach and claims a 99.8% performance on the Brown’s Corpus. Reynar and Ratnaparkhi (1997) 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.

This method can be employed for other languages albeit with minor language specific changes. However, currently it has been tried and tested only on Kannada Texts.

In handling the ambiguity of period in this paper, we consider 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 an abbreviation. 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 such words was obtained from the 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.

Drawing upon the work by Trosterud et. al. (2004), we categorized abbreviations into three classes for the purpose of our algorithm based on whether they are able to end the sentence or not.

a) TRAB: Abbreviations that take an object: (never end the sentence).

Kannada Text: ಮಿ. ಹರೀಶ್.

Transliteration: mi. harIsh.

Translation: Mr. Harish.

b) ITRAB: Intransitive abbreviations (ITRAB) 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.

Kannada Text: 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.

a)AMAB: Abbreviations which are ambiguous- a word is homonymous to an abbreviation.

Kannada Example where a word (verb) may be homonymous to an abbreviation: “ತಾ.”/tA., where “ತಾ.”/tA., could be the verb meaning “bring” or “ತಾ.”/tA., could be an abbreviation for “ತಾರೀಖು”/ tArIkhu meaning “date”.

Kannada Text: ಅದನಿಲ್ಲಿ ತಾ.

Transliteration: adannilli tA.

Translation: Bring that here.

Kannada Text: ತಾ. ೧೫-೦೮-೧೯೪೭

Transliteration: tA. 15-08-1947
Translation:
Date. 15-08-1947

The Algorithm uses 2 word lists as resources, namely ‘valid-sentence-ending-word-list (L1)’ and ‘ambiguous-word-list (L2)’ extracted from the Kannada corpus. This algorithm will disambiguate a period ending token as sentence ending word or abbreviation based on the token length and presence of other dots in the token and matching them with the wordlists.

L1- will have words (tokens) of a language of word-length below a threshold.
L2- will have words (tokens) of a language of word-length below a threshold and homonymous to an abbreviation of that language.

Both L1 and L2 are extracted from the corpus, creating a small set of words.
In this paper, word-length refers to the Unicode Characters, not the count of aksharas.

Algorithm to Identify Period as Sentence Boundary:The algorithm proposed for identifying period as sentence boundary takes as input a continuous text. The Algorithm includes the following steps:
1. Preprocess the text in order to remove any space between a period (“.”) and it’s previous word.
3. Tokenize the text.
4. Group the tokens till a sentence ending marked token is found, such that,

Case1: If the Sentence end marker is either a question mark “?” or exclamation mark “!” then, declare it as the end of sentence.

If the sentence end marker is a period “.” Then
Case2: if the token is numbered token (or if the previous character to the period is a number) then declare the sentence end. This resolves the issue of identifying number as a sentence end with token length less than threshold.

Such as; ರೂ.೧೦. /rU. 10. (Rs. 10) or sentence ending with date such as 20.07.2009.

Case 3: If the token length is less than threshold (In case of Kannada it is 5), then check the token with the open set of 'valid sentence- ending-word-list (L1)' and open set of 'ambiguous-word-list (L2)' for a match.

If the match is found in L1, declare the end of sentence and segment. This solves the issue of tokens with character length less than threshold which are not abbreviations. e.g. 'ನೀನಿಲ್ಲಿ ಬಾ.' / nInilli bA. (You come here).

If a match is found in L2, prompt the user with that word with the context, to allow user to make the decision. This helps to solve the AMAB Issues, by user consent. However, if no match is found in L1, then it's not a sentence end (Possibly initial or salutation). This solves the TRAB issue like 'ಮಿ. ಹರೀಶ್'/ mi. suresha. (Mr. Suresh)

Case 4: If the token length is more than threshold
Check for other possible dots in the token.

- If no other dots are found then declare the sentence end.
- if other dots are found in the token, count the number of characters between the ending dot and its previous dot. If the count is more than threshold, then, declare the sentence end.

This resolves the ITRAB issue like
ತಮ್ಮ ಮೊಟ್ಟಮೊದಲಿನ ನಾಟಕವನ್ನು ಅ.ನ.ಕೃ. ೧೯೨೪ರಲ್ಲಿ ಬರೆದರು.
tamma moTTamodalina nATakavannu a.na.kx. 1924ralli baredaru.
(Aa.Na.Kru. wrote his first drama in 1924.)

If the difference between the two dots in the token is less than threshold then it is not a sentence end, most likely it is an Abbreviation. Don't declare a sentence end. Evaluation:An evaluation of the algorithm 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.

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.

Conclusion: Only 190 such words were extracted from Kannada corpus as 'valid-sentence-ending-words', and not a single word for 'ambiguous-word-list'. Since the check list used in the algorithm is open, it facilitates users to add more words to the list. So as native speakers we were able to identify and provide a small list of ambiguous words. The corpus used for the testing purpose was mainly from two domains – newspaper and literature.

This method can be employed for other languages, however, depending on the language the length of the check lists may vary, as also the threshold.

Sample of the list of valid sentence ending words for Kannada (L1), having frequency greater than or equal to 5.

Frequency	Kannada	Transliteration
910	ಇದೆ.	ide.
352	ನಿಜ.	nija.
81	ರನ್.	ran.
80	ಬೇಡ.	bEDa.
78	ಇವೆ.	ive.
66	ಸಹಜ.	sahaja.
65	ಇದು.	idu.
64	ಸರಿ.	sari.
37	ಅದು.	adu.
34	ಕೂಡ.	kUDa.
29	ಅದೆ.	ade.
26	ವಾದ.	vAda.
24	ದಿನ.	dina.
23	ಅಂತ.	aMta.
23	ಆಸೆ.	Ase.
22	ಅಂಶ.	aMsha.
18	ಶತಕ.	shataka.
16	ಸಲ.	sala.

15	ಔಟ್.	auT.
13	ಏನೋ.	EnO.
13	ಜಯ.	jaya.
13	ತಂಡ.	taMDa.
11	ಆಟ.	ATa.
11	ಸಮಯ.	samaya.
10	ಆಶಯ.	Ashaya.
10	ಎಂದ.	eMda.
10	ಜನ.	jana.
7	ಇರಿ.	iri.
7	ಕಥೆ.	kathe.
7	ಭಾಗ.	bhAga.
5	ಭಯ.	bhaya.
5	ರೋಗ.	rOga.
5	ದೇಶ.	dEsha.

References:

Hla Hla Htay,G.,etal.2006.Constructing English-Myanmar Parallel Corpora. In *Proceedings of ICCA 2006 International Conference on Computer Applications*.Yangon, Myanmar.231-238

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

Riley, Michael D.1989.Some Applications of Tree-Based Modeling to Speech and Language.In *DARPA, Speech and Language Technology Workshop.* Cape Cod, Massachusetts.339-352.

Trond,Trosterud.2004. Preprocessor for Sámi Language Tools. *The Norwegian Sami Parliament, 2004.* Available at www.divvun.no/doc/ling/preprocessor.html

❑❑❑