TrieIR: Indexing and Retrieval Engine for Kannada Unicode Text*

Sumant Kulkarni** and Srinath Srinivasa

International Institute of Information Technology, Bangalore, 26/C, Electronic City, Bangalore, India
{sumant, sri}@iiitb.ac.in

Abstract. Kannada is a phonetic language. In Kannada language, the morphological forms of terms (especially of nouns and verbs) are formed by adding different morphological suffixes to their pure forms. Hence, when queried for morphological forms, search engines based on exact matching fail to identify other semantically similar and morphologically different terms, and thus reduce the quality of the search results. We observe that even though the morphological forms of a term look different, they can be grouped together based on their common prefixes. In this work we propose fuzzy matching based indexing and retrieval algorithms. We propose an indexing mechanism inspired from prefix trees. We also derive our inspirations from the fact that the Unicode encodes the Kannada terms very similar to the way terms are generated using Kannada grammar. We also discuss a query term truncation and decayed score based retrieval algorithm for better retrieval of the documents for the given query. The indexing and retrieval systems still are based on the tf-idf based indexing and retrieval. However, the novelty of the work lies in the way the algorithms bring together the similar terms. This solution can be scaled to work for other South Indian languages with no or little modification as their Unicode encoding and morphological behaviors are similar to Kannada.

1 Introduction

Kannada is the Dravidian style language spoken in the South Indian state of Karnataka. According to 2001 census data, Kannada language is spoken by more than 38 million\(^1\) people. There have been many portals\(^2\) which host large amounts of the Kannada content. However, we argue that this data is being underutilized due to the unavailability of efficient indexing and search methodologies.

---

* This work is a part of Kanaja project, conceptualised by Karnataka Jnana Ayoga.
** The authors would like to thank Prof. S. Rajagopalan and Aditya Ramana Rachakonda for their valuable inputs.

Kannada is a phonetic agglutinative language [5]. In Kannada, both verbs and nouns have morphological form (MF) s which look different than the original term as well as the other MFs of the same term. Hence, when we use the strict term search for Kannada language, which matches only the given term, we can not match the other MFs. Even if the MFs are present, they will not be captured in quantities like term frequency (tf) [6]. This reduces the quality of the search results. Hence, a strict indexing and retrieval methodology, where only the given term is indexed independent of its morphologically related terms, can not work for Kannada text. In this paper, we propose a new indexing and retrieval mechanism for Kannada Unicode text. The indexing is inspired from trie (prefix tree), while the retrieval is done using query term truncate and decay score model.

There have been earlier attempts to understand non-English language IR [2][3][4][5]. A detailed related literature and our approach is available in the technical report [1]. In the further section (2) we discuss our approach and evaluation. We conclude in section (3).

2 Indexing the Document Corpus

In indexing, our main aim is to bring together the different MFs of root term. As discussed, MFs of a term have a single common shorter prefix. However, there can be subsets of MFs having longer common prefixes. This characteristic feature of MFs inspired us to develop a trie based indexing mechanism.

A text corpus contains many documents. Each text document is made up of sequence of words. We parse each document in the corpus individually in to a bag of words. Then, each word is segmented as a set of core-letter and its dependent vowel. For example, the term beTTagaLu is segmented as ba + e + TT a + ga + La + u. The Unicode encoding supports this segmentation. The number of these segments is the length of the term. Hence, the length of the term beTTagaLu is 6. Now on, whenever we mention length in the paper, we refer to this segment length.

Next, for each term we truncate these segments iteratively one by one from the back. Truncation terminates when the term reduces by a truncation factor (α) or the term length reduces below the termination threshold (tt). Here, 0 ≤ α ≤ 1 and and tt ∈ N^0 . The algorithm [1] explains this prefix generation step for a given term. For each document d_i in the corpus, all the prefixes generated are added to bag of prefixes (B_{d_i}). Hence, B_{d_i} is a super-set of the bag of words.

In the next step, called metric generation, we generate prefix frequency (pf) and inverse document frequency (idf). The prefix frequency of prefix p in document d_i is defined as pf_{p,d_i} = log(f_{p,d_i} + 1), where, f_{p,d_i} is the count of p in B_{d_i}. The inverse document frequency for the prefix p is calculated as idf_{p} = log \frac{N}{|D_p|}, where, N is the number of documents in the corpus and |D_p| is the number of bag of prefixes containing p in the corpus. A postings list is maintained to map documents with their pf and idfs. This part of indexing is inspired by classical tf-idf indexing. However, the novelty of the approach is in the prefixes based indexing which enables the retrieval of documents with other morphologies.