Development of Reading Comprehension System for Kannada Text Documents

Anagha Vembar, DivyaTantri, Akshita Saxena, Abhishek Narayanan, Jagadish S Kallimani

Abstract*—Reading Comprehension (RC) plays an important role in Natural Language Processing (NLP) as it reads and understands text written in Natural Language. Reading Comprehension systems comprehend the given document and answer questions in the context of the given document. This paper proposes a Reading Comprehension System for Kannada documents. The RC system analyses text in the Kannada script and allows users to pose questions to It in Kannada. This system is aimed at masses whose primary language is Kannada - who would otherwise have difficulties in parsing through vast Kannada documents for the information they require. This paper discusses the proposed model built using Term Frequency - Inverse Document Frequency (TF-IDF) and its performance in extracting the answers from the context document. The proposed model captures the grammatical structure of Kannada to provide the most accurate answers to the user.

***Keywords*—Reading Comprehension, Natural Language Processing, Kannada, Term Frequency- Inverse Document Frequency, Answer Extraction**

1. INTRODUCTION

Natural Language Processing (NLP) provides machines with the ability to understand human language as it is spoken. NLP has been a widely researched topic with tremendous advances in the English Language. It has been used to perform tasks like Sentiment Analysis, Language Identification, Automatic Summarization, Machine Reading Comprehension and several other tasks.Reading Comprehension is one of the categories that falls under NLP. It is the computational ability of a program to understand text and answer questions based on the context of the text.

With the rise in the volume and the availability of data, there has been an increasing need to apply NLP techniques to comprehend languages other than English. Kannada is one of the most commonly spoken languages in South India. However, as Kannada is a low resource language, there have been comparatively lesser advances in the NLP techniques used for Kannada. It is low resource in the sense that there are very few Kannada datasets available online. In order to work with Kannada documents and apply NLP techniques, one must go through the extensive process of collecting data by themselves. For this reason, Kannada is still a relatively less explored language for NLP and lacks the tools that are available for the English language. A language like Kannada faces a lot of problems such as lack of capitalization, lack of standardized spellings, lack of annotated data, difference in the grammatical structure in comparison with English, scarcity of resources and tools.

The main goal of the Reading Comprehension system is to improve the process of retrieving information from Kannada documents. This could prove to be highly useful for people whose primary language is Kannada.The RC System is able to decipher the information in Kannada documents and answer questions based on these documents.

This paper discusses the proposed model for a RC System - which is based on Term Frequency / Inverse Document Frequency (TF/IDF). The model is tested on pre annotated data to measure its accuracy in providing the answers. The rest of the paper will be arranged as follows: Section 2 discusses the papers and articles that were surveyed. Section 3 explores the methodology of the proposed system and the technical considerations for the selected model. Section 4 discusses the details regarding the implementation of the models. Section 5 analyses the results that were obtained, and Section 6 provides the inferences and concluding remarks. The final section - Section 7 lists out all the references.

2. RELATED WORK

2.1 Reading Comprehension

The paper [1] discusses the creation of the widely used Stanford Question Answering Dataset (SQUAD) and the methods that were used to collect and annotate their data. The data was collected and annotated with the help of crowd workers. The different types of reasoning that would have to be used to answer questions were analyzed. A Logistic Regression model was trained on the data that was collected which yielded good results but was still lacking compared to human comprehension. This dataset lead to a great deal of research in Reading Comprehension.

The paper [3] discusses a new method of creating a Reading Comprehension system that is not limited to basic lexical pattern matching between passages and answers. The main goal of the paper is to create a Reading Comprehension System that uses reference resolution, multiple steps of reasoning, and also uses world knowledge.

Revised Manuscript Received on April 12, 2019.

AnaghaVembar, Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, Bangalore, India. (E-mail: anagha97@gmail.com)

DivyaTantri, Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, Bangalore, India. (E-mail: divya.tantri@gmail.com)

AkshitaSaxena, Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, Bangalore, India. (E-mail: akshitas1503@gmail.com)

Abhishek Narayanan,Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, Bangalore, India. (E-mail: abhisheknarayan004@gmail.com)

Jagadish S Kallimani, Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, Bangalore, India. (E-mail: jagadish.k@msrit.edu)

This leads to a global understanding rather than a sentence level understanding. The paper tests their method on existing neural models which already perform well on the SQUAD dataset. The results show that the existing models show subpar performance on the dataset manually created for this process - which indicates their inability to generalize over different language styles. However, the model developed in this paper performs better than the existing models which shows the feasibility of the proposed method.

The paper [18] discusses Reading Comprehension for multiple paragraphs. The proposed method uses the TFIDF values to determine which paragraph to choose for each particular answer by calculating the similarity between the TFIDF of the paragraph and the question. The paragraphs with the least cosine distance to the question are chosen. The chosen paragraphs are then used to train the Bidirectional GRU. They also proposed another method where all the paragraphs were shared as the same context. In general, the results showed that the shared approach performed better than the TFIDF approach as the model was exposed to more irrelevant details and knew how to choose the correct answer better.

The paper [19] discusses the problem of answering questions without a given context. It consists of two main modules - The Document Retriever and the Document Reader. The Document Retriever is a simple web scraping module that uses TFIDF weighted bag of word vectors to compare the question and the information on the web (Wikipedia). The local word ordering is improved by taking n-gram features into consideration. The next module is the Document Reader which consists of the RNN model that is trained on these collected paragraphs, questions and answers. Wikipedia was used as the knowledge base for all the questions. The Document Retriever was shown to perform well, however the entire system as a whole lacked in performance in comparison to the same RNN on SQUAD dataset. This could be due to the specific paragraphs given in the SQUAD dataset that eliminates any ambiguity.

2.2 Natural Language Processing on Regional Languages

There has been a rise in the research in the NLP in regional languages. This field faces several difficulties such as the difference between the styles of each language and the grammar structures. Generalizing NLP techniques would yield poor results. Several researchers have come up with ways to overcome the difficulties and improve the accuracy.

The paper [9] describes the difficulties of developing grammar productions for South Indian languages. Kannada - being one of the most commonly spoken Dravidian languages - also faces these difficulties. It is lacking in terms of computational linguistics in comparison to other languages such as Telugu. This paper develops a method to tag the Parts of speech in Kannada statements by the use of the Supervised Learning method - Decision Trees, to assign the appropriate POS Tags. The results obtained show the differences between the English grammar structure which is Subject Verb Object (SVO) in contrast with Kannada which follows the Subject Object Verb (SOV) structure.

3. PROPOSED METHOD

In this paper, a model based on Term Frequency – Inverse Document Frequency (TFIDF) is proposed. The model trains on Kannada documents and uses the TFIDF weights to find the answer to the question posed. The TFIDF model was chosen as it can fit very well on a limited amount of data. It is also an unsupervised learning method, which is suitable for our data as there are no publicly available pre annotated or labeled datasets in Kannada.

Figure 1 shows the high-level design of the proposed Reading Comprehension System.

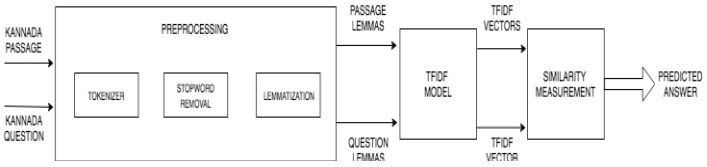

Fig.1. Working of the proposed Reading Comprehension System

3.1 Data Collection

Due to the lack of labelled and annotated data in Kannada, we had to manually find Kannada extracts from different sources and annotate them to test the proposed model. To prevent the model from suffering from annotation bias, the questions and answers were framed by different people. Similar to the collection of data in [1] and [3], the Kannada extracts were distributed to crowd source workers and they framed the questions and annotated the answers.

The Kannada extracts were not limited to a single source and were mainly collected from Kannada Wikipedia and a Kannada newspaper called Vijay Karnataka. Data was collected from articles of different domains to test the ability of the model to infer the correct answers from the extracts given. 50 samples consisting of questions and answers were created and compiled from all the data received by the crowd workers.

Figure 2 describes the procedure followed to annotate the data.

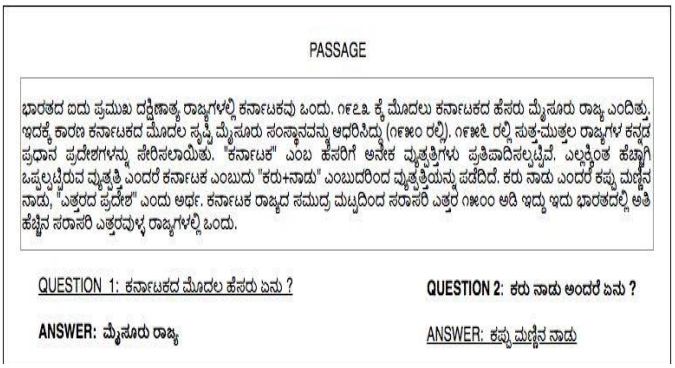

Fig.2. The Underlined text depicts the question and answer framing of one person and the Bold text depicts the question and answer framing of the other person. This prevents annotation bias.

3.2 Preprocessing the Data

The data is preprocessed before providing it as input to the TFIDF model. This step is necessary to obtain the most accurate results.

Tokenization: The input data in its raw form is made up of multiple sentences. These sentences are broken down to their constituent words which are known as tokens.

Stop Word Removal: The words such as "ಮತ್ತು" and "ಅಥವಾ" have very little significance to the overall meaning of the sentences. These words are therefore removed from the passages by finding the inverse document frequencies (IDF) of each of the words. Low IDF values indicate that the word is insignificant and can be removed.

Removal of Punctuation, Symbols and Numbers: The tokens need to further be cleaned by removing all the symbols and numbers as they hold no value in finding document similarities.

Lemmatization: The tokens are now void of most of the unnecessary information. These tokens are now lemmatized, i.e., each token is converted to its root word or lemma. This is performed to prevent any ambiguity while dealing with different forms of the same word.

Once the data has been preprocessed it can serve as the input to the TF/IDF model.

3.3 Term Frequency- Inverse Document Frequency Weighting

The proposed model uses TF/IDF to assign weights to the sentences. It is used to assign scores to words. These scores indicate the importance of a word in the document it occurs in, with respect to an entire corpus of documents. The general formula of TF/IDF is given by Equation (1) below.

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D) \quad (1)$$

where*t* denotes the terms, *d* denotes each document and D denotes the corpus of documents. *tf(t,d)* denotes the term frequency and *idf(t,D)* denotes the inverse document frequency.

The term frequency *tf(t,d)* is the total number of occurrences of term *t* in document *d*.

The inverse document frequency *idf(t,D)* is given by Equation (2).

$$idf(t,D) = log\frac{|D|}{1+|\{d \in D : t \in d\}|} \quad (2)$$

where D refers to the corpus of documents where D = d_1, d_2,…,d_n

This model is applied to each of the sentences in the passage as well as the question. As a result of this, TFIDF scores get generated for each word in the considered sentences. The words/tokens of the sentences could be used individually (unigrams) to find the TFIDF scores or n tokens (n-grams) could be grouped together at a time to find the TFIDF scores. Once these scores are calculated for the n-grams that constitute each sentence, the TFIDF scores are grouped together by sentence. This generates the TFIDF vector that will be used later to find the answer to the question posed.

3.4 Similarity Measurements

Once the TFIDF vectors are generated for each sentence, the final step involves extracting the sentence which contains the answer to the question. Each of the sentence vectors are compared with the question vector in order to find the sentence that is the most similar to the question. The sentence with the highest similarity to the question is predicted to be the answer. This paper uses similarity measures such as Cosine Similarity and Jacquard Similarity to calculate the similarity between the question and the sentences constituting the passage.

4. IMPLEMENTATION

The following pseudo code describes the overview of the proposed method:

```
Input: Kannada passage, Kannada question
Output: Answer in Kannada

Function preprocess (Sentence):
    tokens<- Perform Tokenization on Sentence
    cleanedTokens<- Perform Stop word Removal and
    lemmas<- Perform Lemmatization on cleanedTokens
    return lemmas
Function TFIDF(list):
    for each sentence in list:
        for each token in sentence:
            tfidf<- Find TFIDF of token in sentence
        tfidfVector<- tfidf
    returntfidfVector

Passage  <- Store input Kannada passage in the form of
list of sentences.

Question <- Store input Kannada question.

For each sentence in Passage:
    Lemmas          <- call preprocess(sentence)
    PreprocessedPassage<- Append the lemmas in the
PreprocessedQuestion<- call preprocess(question)
FinalInput<- Append PreprocessedPassage
FinalInput<- Append PreprocessedQuestion
TFIDFVectors<- call TFIDF(FinalInput)
TFIDFQuestion<- Last element of TFIDFVectors
TFIDFPassage<- All elements of TFIDFVectors apart from
the last element

for each vector in TFIDFPassage:
    similarities<- Append similarity between vector and
TFIDFQuestion

AnswerIndex<- Index of the highest element in similarities
Answer    <- Sentence in KannadaPassage at the index -    Answer
```

5. RESULTS

The model was tested on 50 samples that were annotated by different people to help prevent annotation bias. The passages collected were from different domains such as health, geography, politics, etc., Each sample consists of a passage and a question.

In order to perform the preprocessing of the data, tools from NLTK as well as the Shallow Parser created by LTRC, IIITH.

Table 1.Accuracies of the models in terms of the correctly predicted answers:

Model	Accuracy
TFIDF Weighting with Cosine Similarity	77.5 %
Jaccard Index	47.5%
Latent Dirichlet Allocation	35 %

As seen from the table, TFIDF weighting yielded the highest accuracy. The questions it was unable to answer were mainly the questions rephrased using different words than those used in the passages. Since this model relies heavily on term frequency, the accuracy still depends on lexical similarities. It cannot take global understanding of sentences into consideration. However, it was observed to outperform the other techniques by a large margin.

Jaccard Index also works based on common keywords in sentences by taking an intersection over the union of two sets.

Latent Dirichlet Allocation has a global understanding by creating topics from the passages. However, LDA requires a huge amount of data to assign the relevant topics to questions and hence yielded very poor results.

6. CONCLUSION

The proposed model was found to be effective and yielded very good results. However, due to its dependence on lexical similarity, it may not be effective for questions phrased in multiple different ways. In the future, this model could be improved by adding functions to find the exact span of the answers rather than returning the entire sentence in which the answer is present. If more data is collected, supervised learning methods and Neural Networks could be used to yield highly accurate answers by using reasoning and world knowledge.

REFERENCES

1. RajpurkarPranav, Jian Zhang, Konstantin Lopyrev and Percy Liang. “SQuAD: 100, 000+ Questions for Machine Comprehension of Text.” EMNLP (2016).
2. P. P. Walke and S. Karale, "Implementation approaches for various categories of question answering system," 2013 IEEE Conference on Information & Communication Technologies, Thuckalay, Tamil Nadu, India, 2013, pp. 402-407.
3. Wadhwa, Soumya, VarshaEmbar, Matthias Grabmair and Eric Nyberg. “Towards Inference-Oriented Reading Comprehension: ParallelQA.” CoRR abs/1805.03830 (2018)
4. Mittal, Sparsh& Gupta, Saket& Mittal, Ankush. (2008). BioinQAMultidocument Question Answering System: Providing Access to E-learning for masses. Journal of Engineering Students.
5. Stalin, Shalini, Rajeev Pandey and RajuBarskar. “Web Based Application for Hindi Question Answering System.” (2012). nternational Journal of Electronics and Computer Science Engineering. 2. 72-78.
6. Weston, Jason, Antoine Bordes, Sumit Chopra and Tomas Mikolov. “Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks.” CoRR abs/1502.05698 (2016)
7. Pingali, Prasad &Varma, Vasudeva. (2006). Hindi and Telugu to English Cross Language Information Retrieval at CLEF 2006. 1172.
8. Ekbal, Asif&Bandyopadhyay, Sivaji. (2007). A Hidden Markov Model Based Named Entity Recognition System: Bengali and Hindi as Case Studies. 545-552. 10.1007/978-3-540-77046-6_67.
9. Mallamma V Reddy1, Dr. M. Hanumanthappa 2012 POS Tagger for Kannada Sentence Translation. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 1, Issue 1, May-June 2012.
10. Singh, Satyendr&Siddiqui, T.J.. (2015). Utilizing Corpus Statistics for Hindi Word Sense Disambiguation.IAJIT
11. S. Vijay, V. Rai, S. Gupta, A. Vijayvargia and D. M. Sharma, "Extractive text summarisation in hindi," 2017 International Conference on Asian Language Processing (IALP), Singapore, 2017, pp. 318-321.
12. P. Kumar, S. Kashyap, A. Mittal and S. Gupta, "A Hindi Question Answering system for E-learning documents," 2005 3rd International Conference on Intelligent Sensing and Information Processing, Bangalore, 2005, pp. 80-8
13. G. Nanda, M. Dua and K. Singla, "A Hindi Question Answering System using Machine Learning approach," 2016 International Conference *on Computational Techniques in Information and Communication Technologies (*ICCTICT*)*, New Delhi, 2016, pp. 311-314.
14. Sharma, Lovely &Dhir, Vijay &Kaur, Kamaljeet. (2015). A New Model for Question-Answer based Dialogue System for Indian Railways in Hindi Language. Indian Journal of Science and Technology. 8. 10.17485/ijst/2015/v8i35/85941.
15. Moro, Sérgio& Cortez, Paulo & Rita, Paulo. (2015). Business intelligence in banking: A literature analysis from 2002 to 2013 using text mining and latent Dirichlet allocation. Expert Systems with Applications. 42. 1314–1324. 10.1016/j.eswa.2014.09.024.
16. S. Chauhan and P. Chauhan, "Music mood classification based on lyrical analysis of Hindi songs using Latent Dirichlet Allocation," 2016 International Conference on Information Technology (InCITe) - The Next Generation IT Summit on the Theme - Internet of Things: Connect your Worlds, Noida, 2016, pp. 72-76.
17. Gupta, Poonam and Vishal Gupta. “A Survey of Text Question Answering Techniques.” (2012).
18. Clark, Christopher and Matt Gardner. “Simple and Effective Multi-Paragraph Reading Comprehension.” ACL (2018).
19. Chen, Danqi, Adam Fisch, Jason Weston and Antoine Bordes. “Reading Wikipedia to Answer Open-Domain Questions.” ACL (2017).