"Mahoshadha", the Sinhala Tagged Corpus Based Question Answering System

J.A.T.K. Jayakody, T.S.K. Gamlath, W.A.N. Lasantha, K.M.K.P. Premachandra, A. Nugaliyadde and Y. Mallawarachchi

Abstract "Mahoshadha" the Sinhala Question Answering Systems aims at retrieving precise information from a large Sinhala tagged corpus. This paper describes a novel architecture for a Question Answering System which summarizes a tagged corpus and uses the summarization to generate the answers for a query. The summarized corpuses are categorized according to a set of topics enabling fast search for information. K-Nearest Neighbor Algorithms is used in order to cluster the summarized corpuses. The query will be tagged, the tagged query will be used to get more accurate results. Through the tagged query the question will be identified clearly with the category of the query. Support Vector Machine is used in order to both automate the summarization and question understanding. This will enable "Mahoshadha" to answer any type of query as well as summarize any type of Sinhala corpus. This enables the Question Answering System to be more useable through many applications.

Keywords Question answering · Document summarization · Document categorization · SVM algorithm · k-NN classification

J.A.T.K. Jayakody · T.S.K. Gamlath · W.A.N. Lasantha (✉) · K.M.K.P. Premachandra · A. Nugaliyadde · Y. Mallawarachchi
Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
e-mail: it12520640@my.sliit.lk; nlasantha22@gmail.com

J.A.T.K. Jayakody
e-mail: it11189640@my.sliit.lk

T.S.K. Gamlath
e-mail: it12096176@my.sliit.lk

K.M.K.P. Premachandra
e-mail: it12065936@my.sliit.lk

A. Nugaliyadde
e-mail: anupiya.n@sliit.lk

Y. Mallawarachchi
e-mail: yashas.m@sliit.lk

S.C. Satapathy and S. Das (eds.), *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 1*, Smart Innovation, Systems and Technologies 50, DOI 10.1007/978-3-319-30933-0_32

1 Introduction

People are always in a quest for information. With the rapid growth of the available information, Question Answering (QA) systems are mandatory for areas ranging from medical science to personal assistants.

QA differs from Information Retrieval (IR) or Information Extraction (IE). IR systems provide a set of documents related to the query, but do not exactly indicate the correct answer for the query. In IR, the relevant documents are obtained by matching the keywords from the query with a set of index terms from the set of documents. QA is the process of extracting the most precise answer to natural language question asked by the user.

"Mahoshadha" is a fully automated free and open source QA system for Sinhala Language, which helps to answer any question within the provided content. Implications of "Mahoshadha" are immense as it can be adopted by using a tagged corpus according to the application of use. Any annotated Sinhala text document is allowed to input to the system. Some applications where this system can be used are as a medical instructor, artificial teacher, self-learning tool and also can be used in call centres. Therefore it has a high business value as well as its technological service.

"Mahoshadha" is a high research contribution project as it opens a new research area, Question Answering in Sinhala Natural Language Processing.

2 Methodology

The system consists of four components. Document Summarizing, Document categorizing, Question processing and Answer processing.

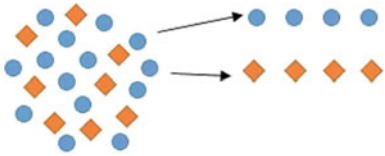
2.1 Document Summarization

"Mahoshadha" has used summarization to summarize the multiple Sinhala documents enter by the user to increase the efficiency by reducing number of terms using Support Vector Machine (SVM) Algorithm. SVM is a Classification method which can be used for Classification and Regression as it supports text mining and pattern recognition.

This uses set of sample documents that have summarized manually as the training documents [1]. As the first step those documents and their respective summaries are provided for the learning process of the algorithm.

After the learning process Classifier (A classifier is a supervised function where the learned attribute is categorical) is used to classify new records (new documents to summarize) by giving them the best target attribute (predicted summary) using text mining and recognizing text patterns as shown in Fig. 1.

Fig. 1 Classification



2.2 Document Categorization

In a QA system, organizing documents in a convenient way is important in order to increase the efficiency of retrieving the answer. In "Mahoshadha" organizing the documents has done by categorizing them to predefined categories considering its content using k-Nearest Neighbor (k-NN) Classification [2]. The method makes use of training documents, which have known categories, and finds the closest neighbors of the new sample document among all [3]. The solution involves a similarity function in finding the confidence of a document to a previously known category.

While categorizing documents, terms that do not have any importance and effect in categorizing documents should be eliminated [4]. Further accuracy can be provided to the categorization as terms added dynamically while adding training documents.

Term Space Model is an important concept in text categorization. First calculate weightages of each terms using the given formulas to use in similarity function [5].

$$wtf(t) = tf * idf(t) \text{ for term t}, \tag{1}$$

$$idf = \log 2(N/n) \tag{2}$$

N Number of all documents
n Number of documents where that term appears

Similarity function takes one training document and the new document as parameter [6]. It returns a value that corresponds to the amount of similarity between these documents.

$$Sim(X, Dj) = [\Sigma ti€(X \cap Dj) xi * dij] / [||X|| * ||Dj||] \tag{3}$$

X New document
Dj Training document *j*
ti term in both vectors
xi *wtf*(*i*)
dij *wtf*(*ij*)

$$||X|| = \sqrt{x1^2 + x2^2 + x3^3 + \cdots} \tag{4}$$

where all x's are wtf of all terms in X. ‖Dj‖ same as ‖X‖ for the terms in training documents (D)

Find the category of new document using k-NN algorithm taking k as the number of training documents.

$$Conf(c,d) = [\Sigma ki€K|(Class(ki') = c)Sim(ki',d)]/[\Sigma ki€KSim(k,d)] \quad (5)$$

(Conf is confidence in long terms.)
c Any category
d New document (X in the formula above)
K Neighborhood of size k for the document X

All similarities between the new document and the documents that belong to class c are added. Then they are divided to all similarities between the new document and training documents belonging to the neighborhood of X in size k [7] (k value used in k-NN algorithm is chosen as the number of training documents). Finally the confidences (conf) are compared and the category, for which the greatest confidence is calculated, is chosen as the category for the new document d (or X).

Using the same method "Mahoshadha" identify the category of the query asked by the user and it will search for the answer in the document belong to that particular category. If the answer couldn't find in that category "Mahoshadha" can search it in the category with next greatest confidence. Likewise the system is capable of finding the answers category wise as a solution for increasing the efficiency.

2.3 Question Processing

This module is the entry point for the project. Under this module discuss about how process on the question which entered to the system and how this module help for the project success. Question processing contain two key components.

- Analyze the Question

  POS tagging
  Language model

- Identify Answer Type (Fig. 2)

Analyze the Question. Inserted question need to be analyzed. It is handled by this component. When analyzing a sentence or set of words first need to come up with a good knowledge about the language. Appearance of the language, behavior of the language, grammar rules are key areas when analyzing a sentence.

POS Tagging. Part of Speech (POS) Tagging is an important process of Natural Language Processing (NLP) and a prerequisite to many other NLP activities. Automatic POS tagging process identifies the syntactic category of each word in a given sentence according to the context where the word of that sentence. POS

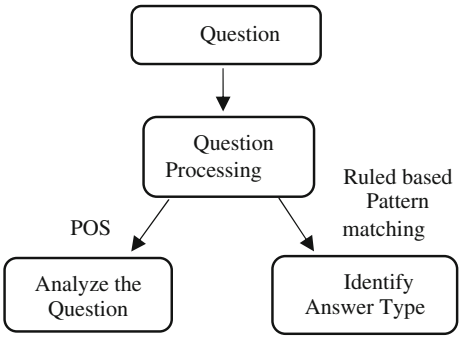


Fig. 2 Question processing

tagger assign the weightages to the each word in dynamically changing context. Tagger cannot process by its own. Need to consider the behavior of the language. (In Sinhala) in this research the POS tagger has been developed using tagged Sinhala corpus of university of Colombo school of computing in Sri Lanka (Fig. 3).

Language model. Language model has responsible to inform about the language to the POS tagger. Language model describe Appearance of the language, behavior of the language, grammar rules of the Sinhala language.

Identify Answer Type. Answer type is a one of a key input to answer processing component. Using answer type, extract answer form selected line in summarized document while answer processing [8]. Many approaches available for answer type identification. For "Mahoshada" project rule based with pattern matching is used to identify the question type.

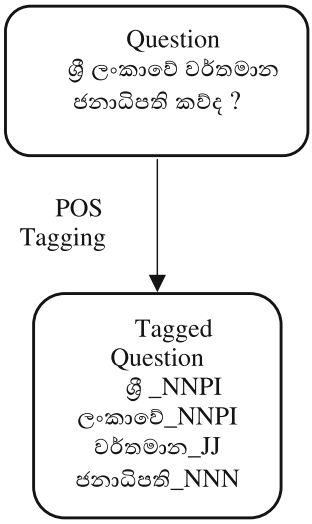


Fig. 3 POS-tagging

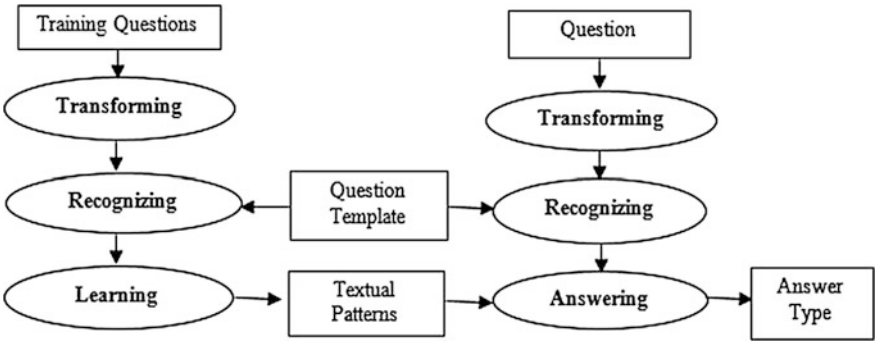


Fig. 4 Architecture of answer type identification using SVM algorithm

Example.
ශ්‍රී ලංකාවේ වර්තමාන ජනාධිපති කවිද ?
Answer type = Person Type
සමාධි පිළිමය පිහිටා ඇත්තේ කොහේද?
Answer type = Location Type

Question types thus derived are used to extract and filter answers in order to improve the overall accuracy of question answering system. To enhance this question classification, we use the question informer feature with Support Vector Machines (SVM). In machine learning approach feature selection is an optimization problem that involves choosing an appropriate feature subset.

As shown in Fig. 4, the system entails two main functions, one is learning and the other is answering. For both functions, the question has to be pre-processed by the transforming and recognizing module.

The SVM Algorithm determines the class of the question. This was done by finding the appropriate template of the question. Currently, [9] we have defined question classes, and each class has several templates formulated as regular expressions which indicate the possible appearance of this type questions. Some examples for question classes are as the follow

කවුද? (who) කුමක්ද? (what)
කොහෙද? (where) කීයද ? (how much)
කොපමනද? (how much) කුමන ?(what)

2.4 Answer Processing

The architecture of this module is shown in Fig. 5.

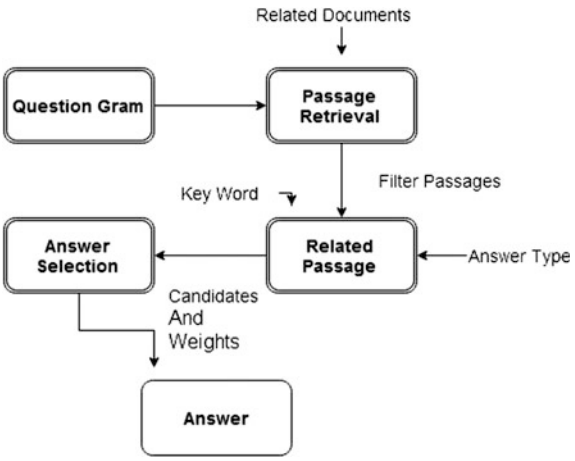


Fig. 5 Architecture of answer processing

The input of this module is constituted by the relevant summarized documents returned by the Question Categorization module, the answer type of the question obtained through the Question Classification module. An n-gram module [10], which we named Question-gram since it creates grams by tokenizing the question, is instantiated for each and every retrieved documents looking for one or several matching passages. This gram starts with a 2-gram and ends with an n (question_lenght-1)-gram. More details about how question-gram works shown in the below (Fig. 6).

There may be more than one passage for a particular question, it may include even unrelated passages also. The process of identifying the most relevant passage is the most crucial process, ability to provide correct answers defend on this process. After retrieving matching passages for a question, those passages goes through a kind of a filter process, end of this process output will be filtered most relevant passage for a particular question. Filter process identifies the passage which has the highest number of question gram.

Example. Suppose for a question there are three passages and highest gram is 4-gram. Then,

Passage 1 contains only 2-gram,
Passage 2 contains only 3-gram and 4-gram,
Passage 3 contains only 2-gram, 3-gram and 4-gram. According to this most relevant passage is passage 3.

This AP module has the relevant passage, keyword identification process starts. This process is to identify an important word for question which helps to identify the most suitable answer. The identification of the answer is done using Positive Point wise Mutual Information (PPMI). Using PPMI equation get the values for each and every question words, simply this tells how many times each word of the question occurred in the retrieved passages, as the key word it generates the

Fig. 6 Example for question gram

question word which has the highest PPMI value [11–13]. Equations used to generate PPMI values shown in the below (6).

$$\text{Word1}_{ppmi} = (\text{Line Frequency} * \text{Question Frequency}) * 100 \tag{6}$$

$$\text{Line Frequency} = \text{Line Count}/\text{Line Size} \tag{7}$$

$$\text{Question Frequency} = \text{Question Count}/\text{Question Size} \tag{8}$$

Finally to get the correct answer keyword and answer type goes through the answer selection process. In this process it retrieves the candidate answers by going through the relevant passage.

Example. If answer type is Person type then it retrieves all the persons in that passage. Then using distance calculation methods it gets the shortest distance between candidate answers and the keyword. As the correct answer it outputs the candidate answer which has the shortest distance.

3 Research Findings/Results and Evidence

"Mahoshadha" is the only attempt of Sinhala QA based on NLP this has become an encouragement for other researchers who are interested in this area. A result of this research, new researches has started for the first time in Sinhala. Best example is a research for QA system for mathematical operations has started as the second part of "Mahoshadha". We have encouraged new researchers to come up with an advanced POS tagger using NLP.

Project "Mahoshadha" is the core application of the research which is based on a Sinhala News corpus. With the success of the core application we have come up with other applications of "Mahoshadha" like "Self Learning Tool for School Children", "Call Assistant Application" for call centres, and "Patient Assistant System" for hospitals that can be very useful in real world.

We provided the self-learning tool to school teachers and students as the first step to get feedback. The product received high positive feedback on finding solution for subject matters themselves, fast and easy to operate the application. Initial request of all of them to provide this application around all the schools island

wide. Therefore we are planning to provide the learning tool for government school free of charge as a social service. Then we will consider about the other applications.

4 Conclusions

The goal of "Mahoshadha" was to retrieving the most accurate answers to the questions ask by users in Sinhala Language. According to the test results the goal has accomplished with 93 % of accuracy and with a high efficiency of generating the answer. "Mahoshadha" opens new paths in Sinhala Natural Language Processing for the researchers as it kept first steps.

Accuracy and efficiency of "Mahoshadha" is increased automatically through machine learning used in summarization, categorization and also in retrieving the answers. "Mahoshadha" can be applied for different types of real world applications.

5 Future Works

We are currently working on delivering the self-learning tool to government schools free of charge as a social service. Using the feedback we hope to improve it further as an answering system for Multiple Choice Questions. This could also be developed as a call assistant system for companies. This could also be used as a Medical Instructor, User Guidance, and Patient Assistant for hospitals and other applications that can be used in mundane activities in order to make our lives easier.

Our next step is to apply "Mahoshadha" for mobiles as mobile application in Android and IOS that can be adapted to any user requirement.

The Application will focus on a voice model which is capable of getting inputs and generating outputs through voice.

Acknowledgment "Mahoshada" team would like to thank Dr. A.R. Weerasingha and Mr. Viraj Welgama of the University of Colombo Language Center for providing a tagged Sinhala News corpus to successfully complete our research.

References

1. Hovy, E.H.: Automated Text Summarization. The Oxford Handbook of Computational Linguistics, pp. 583–598. Oxford University Press, Oxford (2005)
2. Danesh, A., Moshiri, B., Fatemi, O.: Improve text classification accuracy based on classifier fusion methods. In: 10th International Conference on Information Fusion, pp. 1–6 (2007)

3. Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K.: KNN model-based approach in classification. In: Proc. ODBASE, pp. 986–996 (2003)
4. Shin, K., Abraham, A., Han, S.Y.: Improving kNN Text Categorization by Removing Outliers from Training Set, in Computational Linguistics and Intelligent Text Processing, vol. 3878. Series Lecture Notes in Computer Science, pp. 563–566 (2006)
5. Rahal, I., Najadat, H., Perrizo, W.: A P-tree Based K-Nearest Neighbor Text Classifier Using Intervalization. Computer Science Department, North Dakota State University
6. Soucy, P., Mineau, G.W.: A Simple k-NN Algorithm for Text Categorization. Department of Computer Science, Universite Laval, Quebec, Canada, pp. 647–648
7. Miah, M.: Improved k-NN Algorithm for Text Classification. Department of Computer Science and Engineering University of Texas at Arlington, TX, USA, vol. 3, pp. 80–84
8. Tong, S., Koller, D.: Support Vector Machine Active Learning with Applications to Text Classification. Computer Science Department, Stanford University, pp. 287–295 (1998)
9. Cristianini, C., Taylor, J.S.: An Introduction to Support Vector Machine, pp. 206–240. Cambridge University Press, Cambridge (2000)
10. Buscaldi, D., Rosso, P., Gómez-Soriano, J.M., Sanchis, E.: Answering questions with an n-gram based passage retrieval engine. J. Intell. Inf. Sys. **34**, 113–134 (2010)
11. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Comput. Linguist. **16**(1), 22–29 (1990)
12. Turney, P.D., Pantel, P.: From frequency to meaning: vector space models of semantics. J. Artif. Intell. Res. **37**(1), 141–188 (2010)
13. Dagan, I., Pereira, F., Lee, L.: Similarity-based estimation of word co-occurrence probabilities. Mach. Learn. **34**(1), 43–69 (1999)