Keyword Extraction and Clustering for Expressive Opinion Generation

Hrishikesh Kulkarni, Aditya Daptardar, Vishwajeet Patil & Mithilesh Mane
Department Of Computer Engineering, Savitribai Phule Pune University, Pune, Maharashtra, India

Abstract: Humans are surrounded by an unprecedented wealth of information, available as documents, databases, or multimedia resources. Access to this information is conditioned by the availability of searching techniques. Keyword extraction and clustering for expressive opinion generation intelligently generates automatic expressive opinion sentences as a response to the user query based upon the transcript provided. Earliest techniques have used Term Frequency Inverse Document Frequency (TFIDF) values and word concurrence frequencies to rank words for extraction, these approaches do not consider word meaning so they may ignore low frequency words which together indicate a highly salient topic.

The system being developed uses reward functions along with novel algorithmic approach for overcoming the limitations of previous systems thereby resolving the problems of keyword extraction and topic diversity coverage. Here we are using the Stanford parser for Named Entity Recognition. Information extraction is conditioned by the Entities recognized by the system.

The main purpose of the system is to maintain multiple hypothesis about the users information needs and to present a small sample of recommendations based on the most likes once along with the sentiment of the response.

Keywords— Keywords Extraction, frequency, Rank keywords Clustering, Opinion Generation, Sentence Generation, Named Entity Recognition.

1. Introduction

Keyword extraction and clustering for expressive opinion generation [1] [2] is an integration of summarization and natural language querying system. System provides facility to input transcript and then analyze it for the summary and probable user’s queries. System has five basic modules. First module is Text Pre-processing module which helps in analyzing the text for compatibility and removes any noise from the text. Second module is summarization module which uses Naïve Bayes algorithm for text summarization [3]. Third module is Natural Language querying module which analyzes the user query based upon Stanford Named Entity Recognition Classifier. Fourth module is Sentence Generation which consists of relevance searching using subject-object-predicate relation. Fifth module is classification and sentiment analysis which classifies the transcript to a particular news domain, and the sentiment analysis calculate the sentiment of the generated sentence.

2. Design and Implementation of the System

The system being developed considers transcript as its primary input and generates a summarized knowledge base using Naïve Bayes algorithm along with keyword extraction and clustering. In second phase a user query in natural language is taken as secondary input which is internally analyzed and interpreted using Named Entity Recognition Classifier. Which is then executed on the summarized knowledge base to generate expressive opinion.
A. System Architecture

The overall architecture narrates the complete designing of system. It also provides the execution flow and module integration sequence. The input to the system is in text format currently limited to 200 Lines. Text-Preprocessing removes special characters and recognizes all character sequences of English language along with its respective punctuation. In Text summarization for generating knowledge base Naïve Bayes algorithm is used which uses keyword extraction, ranking and clustering. User queries are in natural language which are analyzed using Stanford Named Entity Recognition Classifier. Sentence generation is used for selecting relevant and context based data by dividing the knowledge base in subject-object-predicate pattern [2].

B. Mathematical Equations

Naive Bayes [4] is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set.

Abstractly, naive Bayes is a conditional probability model: given a problem instance to be classified, represented by a vector \( X=(x_1, x_2, \ldots, x_n) \) representing some \( n \) features (independent variables), it assigns to this instance probabilities.

\[
p(C_k|x_1, \ldots, x_n)
\]

For each of \( K \) possible outcomes or classes.

The problem with the above formulation is that if the number of features \( n \) is large or if a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable. Using Bayes' theorem [5], the conditional probability can be decomposed as

\[
p(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)}
\]

C. The Implementation of the System

The System is developed using multiple edge cutting technology like java swing and Stanford Natural Language Parser. The application is developed as single user desktop application. The Pre-processing module is implemented using Document Preprocessing guidelines of Stanford Parser. Summarization using Naïve Bayes is implemented at sentence level [3]. Sentence Generation uses relevance based selection and matching of subject-object-predicate sequence for sentence reconstruction. Classification of domain of news and sentiment analysis is done using Stanford Sentiment Analysis. The System is trained using 20NewsGroups Dataset.

Following observations have been recorded based on the system performance.

Response Time:

10 seconds is about the limit for keeping the user’s attention focused on the dialogue. For longer delays, users will want to perform other tasks while waiting for the computer to finish, so they should be given feedback indicating when the computer expects to be done. Feedback during the delay is especially important if the response time is likely to be highly variable, since users will then not know what to expect.
Work Load:
In execution of a single sentence query, it needs to go through 5 main modules of the System, which approximately takes 1 sec per module, so the graph shows work load as below.

![Workload Graph]

3. Conclusion

Keyword Extraction and Clustering for Expressive Opinion Generation is best suited to be used in systems like Dynamic Intelligent Assistant, Context Based Knowledge Navigator, and Human Computer Interface. Where opinion and relevance to the current context is eminent. The main contribution of this research to the question answering systems research domain is considerations of informative responding, specifically, expressive opinions. Conventional question answering systems are primarily focused on functional interactions to achieve specific tasks; therefore, they are based on Grice’s cooperative conversation principles. However, they are not enough to attract users to engage with the system. In this paper, we assumed informative productions in our daily enjoyable conversations appear as an interlocutor’s original way of expressions and viewpoints, which can be represented as frequency of adjectives. Beyond simple exchange of questions and answers, just like most current academic and industrial question answering systems, the expressive opinions and additional phrasing mechanisms could possibly trigger users’ motivations to continue to interact with systems over a period of time. Our proposed informative question answering system framework has huge potential for QA-typed recommender systems. Misu et al. presented a system-initiative information recommendation method as an application of a question answering system for user satisfaction [6]. They proposed system-initiated spontaneous questions as a recommendation method. Beyond that, a mechanism of our system’s additional expressive opinion has potential to offer serendipitous ideas to users, which might realize more enjoyable conversations. Misu et al. achieved high user satisfaction ratings by making topic-diversified recommendations and reducing the similarity of recommendation lists [6]. Murakami et al. assumed that user satisfaction depends on whether a recommender system suggests unexpected items that are relevant to the users’ preferences [7]. In order to apply our framework for more serendipitous recommender systems, we will consider learning mechanisms of opinion ranking algorithms and sentence combination based on conversational contexts and user models to maximize users’ satisfaction.

4. Acknowledgements

The authors wish to thank the support by Dr. Prof. J.R Prasad (Department of Computer Engineering, Singhad College of Engineering) for this which is nothing but a corollary of her motivation.

5. References


