

Project Report
of
**Question Answering System for e-commerce
sites**

[Guide](#)

Prof. Radhika Mamidi

[Team](#)

Ganesh Borle (201505587)
Satyam Verma(201505604)
Sonam Gupta(201505506)
Vivek Vishnoi(20105504)

Abstract

These days most of the shopping is online with the growth of e-commerce. In this scenario, a customer or a user would want to know about a particular product before buying it . A customer service provider answers questions posed by users going through the product, pricing and personal information of which they have access to. So, our project is to build a question answering system that would answer products-related questions of the users. Through this project we would like to automate the task performed by customer service providers.

Challenges

- The scope for the QnA is very large. Hence we need to fix a domain to narrow it down.
- We will have to fix a window size to keep track of the previous dialogs or queries that the user enter.
- Extracting the aspect term from the query, and ask the user for more information if required.
- Anaphora or coreference resolution.
- Classification of queries into bins based on the type of response like descriptive, yes/no question, etc.
- Ranking the relevant responses.

Project Scope

On a broad level the project aims to take a sentence as an input and extracts the aspect term form it, and tries to use a simple pattern matching algorithm to retrieve the relevant answers from the accessible data.

The Scope can be divided into the following levels:

1. Data representation using appropriate data structures.
2. Aspect Term detection in the query.
3. Identify the right fields in the constructed data structure of the available data.
4. Generate a sentence asking the customer to provide the required information, if anything is missing.
5. keeping track of the information in previous question, so that the context doesn't get lost.
6. Classification of queries into bins based on the type of response like descriptive, yes/no question, etc.
7. Ranking the relevant responses.

Related System :

1. Apple's Siri is something similar to this but far more advanced.
2. [Niki](#) is a AI powered chatbot that lets you search, discover & pay for multiple services.
3. START Natural Language Question Answering System is a system developed at MIT. ([Link](#))

Proposed Approach :

- Removing stop words: Stop words are the words which appear frequently in the query but provide less meaning in identifying the important content of the document such as 'a', 'an', 'the', etc.
 - Stemming: Word stemming is the process of removing prefixes and suffixes of each word.
 - We then identify the tokens which define many of the possible domain questions and answerable token keywords which enable our system to search questions more efficiently. (**Aspect term extraction**)
 - The system will search for the term that is found in *Step 3* and its associative word found in *Step 2*. If the user asks the question: "What is the warranty period for a Videocon TV model: VIDCN 12345".
Then the system will search for the domain 'TV', then sub-domain: 'Videocon' and then for model number that is in the user query.
1. If the user's question is incomplete, then our system will prompt for the missing information.
 2. System will auto generate questions depending on the previous questions. If the user asks for 'television and warranty period' in the previous queries then our system will ask further questions like 'warranty period range' or 'would you interested in screen size', etc.

Tools/Technologies to be used :

1. Scrapy - For crawling and collecting data.
2. Stanford's Aspect Term Tool - For extracting the aspect term from the query.
3. Stanford NLP Parser & POS-Tagger
4. Natural Language Toolkit (NLTK)
5. Python

Future Work :

1. Maintaining the context across the queries (Dialogue System)
2. Handling of more complex queries
3. Scaling of the architecture so that more concurrent queries can be handled
4. Expand to more domains other than mobiles like medical etc.
5. Addition of voice communicator

Links :

Git : <https://github.com/buggynap/QnA>

DropBox : <https://www.dropbox.com/sh/b71qqlej0xle06i/AADbKNqny3A61lgKlsj8lgLWa?dl=0>

References

- Green, W., Chomsky, Laugherty, C. K.(1961). BASEBALL: An automatic question answerer, Proceedings of the Western Joint Computer Conference, pp 219- 224
- Srihari, R. & Li,W. (2000). "Information extraction supported question answering" In Proceedings 8th Text Retrieval Conference (TREC-8), NIST Special Publication 500-246.
- Kim. H. Kim. K. Lee, G.G., & Seo. J. (2001).MAYA: "A Fast Question Answering system based on a predictive answer indexer."