**WEB CRAWLER AND CHATBOT REPORT**

**NLP CS 6320**

**KANYA KRISHI**

**KXK220008**

**INDEX**

**Topic Page.No**

1. Web Crawler
2. Introduction 2
3. Knowledge Base Creation 2 - 3
4. Screen shots of Knowledge Base 3 - 4
5. Important terms 4
6. ChatBot
   1. System Description and NLP techniques used 5 - 6
   2. Dialog Tree or Logic
   3. Sample dialog interactions
   4. Appendix and Live lookup with samples
   5. Sample User Models
   6. Evaluations of Chatbot

**WEB CRAWLER**

**Introduction**

This chatbot has been developed employing Natural Language Processing (NLP) techniques acquired to date. It is designed to engage in focused conversations within a specific domain by leveraging a knowledge base or extracting information directly from the internet. The chosen domain for this application is DOGS. Users may pose questions regarding dog breeds, dog nutrition, the adoption process for dogs, and related topics.

**Knowledge Base Creation**

A comprehensive knowledge base was established and subsequently utilized for reference within the chatbot framework. This involved the formulation of a dictionary containing pertinent facts. The following steps were undertaken:

1. Web Crawling and Content Acquisition: This foundational phase involves constructing a web crawler designed for methodical information extraction from various online sources. The process initiates with the following actions:

* Seed URLs: Establishment of initial entry points for the crawler, personally selected for relevance and depth.
* Discovery of New URLs: Systematic navigation from each seed URL to collect HTML content and identify hyperlinks leading to additional pertinent web pages.
* URL Queue Management: Implementation of a queue system to catalog URLs pending exploration, thereby streamlining the crawling sequence. This approach ensures prioritization of new links and prevents redundant revisits to already processed URLs. A predefined maximum value restricts the total number of URLs to be crawled.

1. Text Cleaning and Preprocessing: The raw HTML content sourced during the crawling phase undergoes preprocessing to facilitate the extraction of significant terms. It includes.

* Text Normalization: This step involves transforming all text to lowercase to maintain uniformity and minimize the impact of capitalization on subsequent analyses.
* Tokenization: This involves breaking down the text into individual words or meaningful units, preparing it for deeper analytical processes.
* Stop Word Removal: This process eliminates commonly used words (such as "the", "a", "an"), which provide limited semantic contribution.
* Lemmatization: This refers to the process of converting words to their base or root forms (for example, transforming "running" to "run"), which helps in standardizing variations and enhancing the consistency of the data.

1. Important term identification: A technique called TF-IDF (Term Frequency – Inverse Document Frequency) is used on the preprocessed text obtained from the previous step. The TF score measure how frequently a particular term appears within a specific document. The IDF score is used to capture the significance of a term across the entire corpus of crawled webpages. Commonly occurring terms have a lower IDF value while those that appear infrequently have a higher value. Hence the product of the TF and IDF values sever as an indicator of a term’s overall importance in the data crawled. The top 30-40 terms are displayed.
2. Utilizing extracted terms : The top important terms which were obtained from the previous step are used as keys in the knowledge base. Corresponding facts have been added to the respective key to form the knowledge base. I used my knowledge about dogs as well as Google search to assist in the final creation of the knowledge base.

**Screenshots of Knowledge base created**

A screenshot of a computer screen

Description automatically generated

**A screenshot of a computer

Description automatically generated**

**Important terms**

The below list displays all the important terms.

1. Important terms extracted using TF-IDF:

TOP TERMS: ['purina' 'breed' 'food' 'article' 'advice' 'names' 'read' 'name' 'product' 'your' 'senior' 'type' 'free' 'puppy' 'dog' 'find' 'feeding' 'where' 'kitten' 'dogs' 'finding' 'what' 'newsletter' 'guides' 'owner' 'adult' 'brand' 'topic' 'plan' 'breeds' 'guide' 'treats' 'getting' 'best' 'care' 'pet' 'black' 'join' 'contact' 'page' 'shop' 'nutrition' 'need' 'right' 'adventurous' 'quiz' 'every' 'online' 'feed' 'help']

1. Manually determining Important terms based on my domain knowledge

IMPORTANT TERMS: breed, training, nutrition, vaccination, socialization, obedience, grooming, spaying, neutering, adoption, behavior, exercise, health, therapy, rescue, pedigree, allergies, temperament, microchipping, leash

**CHAT BOT**

**System description**

This dog-focused chatbot is designed to facilitate constrained interactions with users, meticulously extracting the most suitable responses from its extensive knowledge base through the application of diverse Natural Language Processing (NLP) methodologies. If a query does not correspond with any existing entry within the knowledge base, the system initiates a real-time web search to procure pertinent information. Moreover, for inquiries that remain unresolved, the chatbot offers tailored statements or suggestions related to topics of potential relevance, thereby providing users with constructive guidance, and enhancing the overall user experience.

**NLP techniques incorporated**

Upon receiving a query from the user, the system is engineered to identify and present the most pertinent response available. In order to ascertain the response that most closely aligns with the user's inquiry from the knowledge base, the system employs the following techniques:

1. Part-of-Speech tagging

**Overview**: A core NLP method that assigns categories (e.g., noun, verb, adjective) to words in text. My chatbot employs this technique mainly to isolate nouns from both user queries and knowledge base entries.

**Application**:

1. **Noun Extraction**: The **get\_nouns\_from\_pos(text)** function processes input (queries or facts) through an NLP model to tokenize text and filter out NOUNS for further ananlyis.
2. **Similarity Determination**: The **find\_most\_similar\_pos\_tagging(user\_query)** function extracts nouns from user queries and knowledge base facts, comparing them to identify overlaps. A predefined noun overlap threshold is essential for a fact to qualify as a potential response, ensuring relevance.
3. **Ranking and Selection**: Facts are ranked by noun overlap magnitude, with the top-ranking fact deemed the most suitable response. The match from the ranked list based on user’s likes are given more preference. If there is nothing common, then the first result from the sorted list is displayed.
4. Cosine similarity

**Overview**: A key NLP metric for assessing the similarity between two documents regardless of size. This chatbot employs cosine similarity to compare user queries with existing facts.

**Application**:

1. **Similarity Calculation**: Utilizes TF-IDF vectors through the **calculate\_similarity** function to quantify similarity between a user query and facts.
2. **Threshold-Based Matching**: The **get\_cosine\_similarity** function computes similarity scores, applying a threshold to filter for relevance.
3. **Ranking**: Facts are prioritized by their similarity score, selecting either matches aligned with user preferences or the highest scoring fact.Top of Form

Bottom of Form

1. Dependency Parsing

**Overview**: An NLP technique that analyzes sentence structure to identify grammatical relationships between words. Used in our chatbot to break down both user queries and knowledge base facts into entities and their relational dynamics.

**Application**:

1. **Extraction**: Utilizing **get\_entities\_and\_relations\_from\_dependency\_parsing**, the system identifies crucial entities (nouns, subjects, objects) and their interrelations (adjectives, modifiers, prepositions), within texts.
2. **Matching**: In **find\_most\_similar\_dependency\_parsing** the system will match user queries with facts by evaluating grammatical and semantic overlaps, with a predefined threshold for relevance.
3. **Ranking**: facts are ranked based on the overlap in entities and relations. The selecting the most contextually aligned fact as the response.