NLP WRITE UP FOR QUESTION AND ANSWERING

Task: Question and Answering. Program needs to generate an answer to a given question, as input.

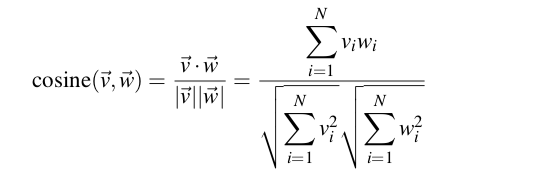
For the task three Natural Language Processing Ideas would be employed to solve them. These ideas include:

1. Cosine Similarity and Information Retrieval Based IR)
2. Naïve Bayes and Rule-Based Approach
3. Latent Dirichlet Allocation (LDA) and Rule-Based Approaches
4. COSINE SIMILARITY AND INFORMATION RETRIEVAL APPROACH (IR)

With this idea, cosine similarity and information retrieval act as components of solving the task where, cosine similarity provides us with information on how to identify questions and IR-Based model provides answers to these questions. The details of how they solve the task are outlined below:

* 1. **What is Cosine Similarity?**

To define similarity between two target words v and w, we need a measure for taking two such vectors and giving a measure of vector similarity. By far the most common similarity metric is the cosine of the angle between the vectors. as: The cosine similarity metric between two vectors ⃗v and ⃗w thus can be computed:



* 1. **Why is it Useful to the Problem?**

In this task, we want to be able to identify a user’s query and its most probable match amongst the questions being trained on. Given a trained question t1, “Who is American’s president? and a user’s query, “Who is the Head of America?”, the similarity between trained question t1 and the user’s query should be higher than another given trained question 2, “What is the length of a Dice?” and thus the user would be most likely referring to trained question, t1 of which we have an answer to. The idea here is we need to be able to identify that two questions may be similar such that we can return the answer of one question to another. With the assumption that questions that are close to being the same, have answers that are close to being the same.

* 1. **What is Information Retrieval Based Approach?**

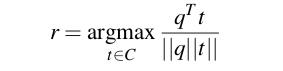
Information retrieval (IR) is the task of finding the document d from the D documents in some collection that best matches a query q. For IR we’ll therefore also represent a query by a vector, also of length |V |, and we’ll need a way to compare two vectors to find how similar they are. (Doing IR will also require efficient ways to store and manipulate these vectors, which is accomplished by making use of the convenient fact that these vectors are sparse, i.e., mostly zeros).

Given the corpus and the user’s sentence, IR-based systems can use any retrieval algorithm to choose an appropriate response from the corpus. The two simplest methods are the following:

1. Return the response to the most similar turn: Given user query q and a conversational corpus C, find the turn t in C that is most similar to q (for example has the highest cosine with q) and return the following turn, i.e. the human response to t in C:



1. Return the most similar turn: Given user query q and a conversational corpus C, return the turn t in C that is most similar to q (for example has the highest cosine with q):



* 1. **Why is it Useful to the Problem?**

Information retrieval is useful in addition to cosine similarity because after identifying that two questions may be the same, we need to retrieve the answer to the trained question in the corpus.

* 1. **Steps for Implementation**

1. Create a corpus of questions and answer pairs
2. For every user query, compare query to question in corpus and use cosine to pick the most similar
3. After finding most similar question, the answer to the question is returned to the user.
4. NAÏVE BAYES TOGETHER WITH RULE-BASED APPROACH (HYRBRID)

With this idea, a Naïve Bayes Model is trained on a corpus of Questions and their corresponding answers. This does not make it enough to answer questions, as questions may be twisted such that they may not exist in our dataset. Thus, the features extracted need to be rich. This is where the Rule-Based comes in.

* 1. **What is Naïve Bayes Classification?**

Naive Bayes Classification is a probabilistic technique used to assign category/class labels to a given input by assuming conditional independence between features. The class labels are drawn from some finite set. It is a form of supervised learning in that it takes a set of inputs and their corresponding output labels trains on them and is able to take a new input X and for the set of output classes Y1 to YN determines if the class of X is any of the output classes. There are three types of Naive Bayes implementations:

1. Gaussian Naive Bayes: Useful if your data follows a normal distribution
2. Multinomial Naive Bayes: Useful when data can be represented as a vector of counts. Works close to the Bernoulli but goes further than binary outputs.
3. Bernoulli Naive Bayes: Useful if your output classes are binary
   1. **Why is it Useful to the Problem?**

For our current task, we have a list of questions and their corresponding answers. The answers can be thought of labels within which the questions belong. Naive Bayes is useful for breaking down the questions into features and determining the probability of a given answer generating a particular feature/question. When given a new question, we would be able to use these probabilities to determine the answer more likely to get to the question based on certain features. However, these features may not be enough to determine the roots of sentences in order to compare similarities. Thus, rules are constructed to generate the roots of the sentences to compare.

* 1. **What is Rule-Based Approaches?**

Rule based underline the general approach of using rules encompassing the structure of source text, syntax, semantics, morphology and roots of source text and output text, to identify a match. In using Rule-Based approaches in the question and answering task, we intend generate pattern rules that are linked to keywords and roots of words which might occur in sentences such that, we can compare if a user query compares to a question in our dataset.

* 1. **Why is it Useful to the Problem?**

It is useful to this problem because we need to be able to make the claim that a given user query looks most similar to a question in our trained dataset, and this can be done be identifying keywords or roots such that sentences that have the similar roots, would most like be similar. These roots are used as features of the system, for the Naïve Bayes to perform classification based on these lists.

* 1. **Steps for Implementation**

Example of Rules Employed includes:

1. Specific words are more highly ranked
2. General words such as “the”, “who” and “why” are ranked lower
3. Punctuations and Normalization patterns

Steps:

1. Example Rules above are used to extract keywords and roots of features sentences
2. These keywords and roots are extracted from the user query and compared to the keywords and features of trained questions
3. The one with the most similar roots with the user’s query is chosen
4. The answer to the question is picked as the answer to the user query by the Naïve Bayes Model

1. LATENT DIRICHLET ALLOCATION (LDA) AND RULE-BASED APPROACHES
   1. **What is Latent Dirichlet Allocation?**

Latent Dirichlet Allocation (LDA) is an example of topic modeling and is used to classify text in a document to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions. What happens with LDA, given a set of documents containing words, it makes a prediction about the categories of the documents based on the number of similar words it sees between documents. It refines its prediction by making more observations about the similar words in documents. After enough observations, the document is able to categorize documents accurately using the grouping of similar words within the documents.

* 1. **Why is it Useful to the Problem?**

For our current task, we wish to answer questions that are frequently asked which could be represented in different formats. The role of LDA should be able to identify and these similarities after Rules have extracted keywords and roots, to label questions with answers. LDA

* 1. **Steps for Implementation**

Example of Rules Employed includes:

1. The lower and upper boundary of the range of n-values for different n-grams to be extracted should be [1,1]
2. When building vocabulary ignore items that have a document frequency strictly lower than the given threshold which is 1

Steps:

1. The Latent Dirichlet Allocation library was imported to enable us to build the LDA model. Also, the Count Vectorizer library was imported to help normalized each sentence in the training data file and to keep counts of the occurrence of the various words in each sentence. Data **normalization** was employed in this step as well.
2. The vectorized data was then transforming to obtain integer representations of each feature per sentence.
3. The LDA model was then built by calling the Latent Dirichlet Allocation class library.
4. A “*most\_similar*” function was then defined to determine the most similar **answer** that corresponds to a question passed as a test sentence