International Journal of Information Retrieval Research
Volume 12 • Issue 1

Why-Type Question to Query Reformulation for Efficient Document Retrieval

Manvi Breja, National Institute of Technology, Kurukshetra, India
https://orcid.org/0000-0003-0607-3094

Sanjay Kumar Jain, National Institute of Technology, Kurukshetra, India
https://orcid.org/0000-0003-1999-5530

ABSTRACT

Understanding the actual need of users from a question is very crucial in non-factoid why-question answering as why-questions are complex and involve ambiguity and redundancy in their understanding. The precise requirement is to determine the focus of question and reformulate them accordingly to retrieve expected answers to a question. The paper analyzes different types of why-questions and proposes an algorithm for each class to determine the focus and reformulate it into a query by appending focal terms and cue phrase 'because' with it. Further, a user interface is implemented which asks input why-question, applies different components of question, reformulates it, and finally, retrieves web pages by posing query to Google search engine. To measure the accuracy of the process, user feedback is taken which asks them to assign scoring from 1 to 10, on how relevant are the retrieved web pages according to their understanding. The results depict that maximum precision of 89% is achieved in informational type why-questions and minimum of 48% in opinionated type why-questions.

KEYWORDS

Constituency Parsing, Dependency Parsing, Document Retrieval, Non-Factoid, Precision, Question Answering System, Question Reformulation, User Feedback

INTRODUCTION

Question Reformulation is one of the components of Question Analysis module in Question Answering System. Question Reformulation reformulates the input question according to user's need in order to affect the accuracy of subsequent modules. Why-type non-factoid questions are complex and ambiguous; making them difficult to answer. It is difficult to understand the actual need of user and derive an appropriate non-ambiguous meaning to it. If a correct query is posed to a search engine, it retrieves appropriate web pages that ultimately help in accurate document retrieval. In English language, there are two broad categorizations of questions (1) Factoid questions of type what, where, which, when and who; (2) Non-Factoid questions of type why and how. The factoid questions are simple and non-ambiguous whereas non-factoid questions are complex and difficult to answer.

Question Reformulation plays a crucial role in question answering system. It retransforms question into an appropriate query that depicts the user's need and thus helps in efficient answer retrieval. The performance of question reformulation affects the performance of subsequent modules, i.e. document, answer candidate extraction and answer re-ranker (Kangavari et al., 2008).

DOI: 10.4018/IJIRR.289948

Query reformulation is a key task in today's web search engines for retrieving accurate and best results corresponding to the users' query. Query reformulation is a process of modifying original query to resolve problems of ambiguity, vocabulary mismatch and vagueness. There are different techniques to query reformulation viz. (1) query expansion, (2) query suggestion and (3) query refinement (Ooi et al., 2015).

Query expansion expands query based on (a) relevance feedback by finding co-occurring terms, (b) query terms appended by their synonyms retrieved from WordNet and (c) retrieved informative terms for expansion from definition clusters (Bernhard, 2010). Query refinement modifies query based on the users' past query logs. It doesn't provide choice to user in selecting terms which can be appended to query. Terms are generated based on user feedback from the top ranked documents irrelevant to its appropriateness which helps in achieving high recall and precision. Finally query suggestion helps understanding the actual information need of user and is found as the most fundamental features of search engines. They are often required in case of rare query being posed, single-term query, unambiguous query suggestions, query suggestions are generalized form of original query and several pages are crawled by user. The approach suggests several other refined query corresponding to original user query based on the users' interest/search logs analysis so that user can select terms that should be replaced original terms for better document retrieval.

The paper focuses on improving the why-question answering system by reformulating why-questions into an appropriate query that can depict the user's need and when posed on search engine, help in retrieving appropriate web pages. There are some cases where the actual user need can't be understood from the question, thus there comes the need for analyzing the question and reformulates it into an appropriate form that can depict the user need from the question.

The organization of paper is described as: Section 2 discusses researches of reformulation. Section 3 puts light on the main focus of the article. Section 4 analyzes different components of question with their impact on reformulation. Section 5 discusses algorithm designed for reformulation of different why-type questions. Section 6 highlights implementation details utilized while designing a user interface for reformulation. Section 7 describes results with their analysis on user feedback. Finally section 8 concludes the work.

BACKGROUND

This section highlights different researches done for improving question answering system utilizing query reformulation and expansion. Kangavari et al. (2008) identified various ways to express answers to a question in question reformulation component. The authors adopted syntactic and semantic relations between words of question, utilizing patterns and other information of previous existing questions which are similar to users' question. Herdagdelen et al. (2010) utilized integrated syntactic and semantic models with Levenshtein distance algorithms for reformulating query which improved the performance in retrieving documents. Pires (2012) developed JustAsk QAS with Query classification and Reformulation to improve passage retrieval by understanding users' information need. They adopted reformulation techniques by designing 13 matching patterns at lexical level. Umamaheswari et al. (2012) utilized semantic based reformulation technique by generating patterns on the basis of lexical, syntactic and semantic constraints. The technique is applied on TREC dataset to retrieve their accurate answer. Each retrieved candidate answer is weighted on the parameters of length, semantic similarity between Question & Answer and distance between keyword to attain a precision of 0.49. Musa et al. (2019) proposed a QAS architecture comprising three modules (a) Rewriter module which reformulates a question into queries by selecting important terms using ConceptNet embeddings, (b) Retriever module which retrieves relevant passages corresponding to queries and (c) Resolver which utilizes textual entailment probabilities to determine the best final answer. Esposito et al. (2020) proposed hybrid query expansion method which extracts synonyms and hypernyms of question terms from MultiWordNet. The resulting set is ranked based on different question words and senses.

Relevant document sentences are retrieved and effectiveness is measured & assessed for candidate answer sentences. Herrara et al. (2021) proposed reformulation of Spanish questions as a component of QAS. Questions are reformulated into new individual questions based on the question elements such as lexical category of each term, named entities and multi-word terms. Further the grammatical elements are identified on different question classes trained on CNN models which further help to locate the question focus in order to rephrase it properly. Vakulenko et al. (2021) discussed question re-writing component for conversational QA. Ambiguous question asked in conversational context are reformulated into unambiguous question. The method was adopted for two tasks i.e. retrieval QA for finding an answer to a question as a ranked list of passages and extractive QA for finding an answer to a question as a text span within a passage. Different Question Reformulation (QR) models are tested where Transformer++ performed best with 0.81 ROUGE score on CANARD dataset and 0.9 ROUGE score on TREC CAsT dataset.

MAIN FOCUS OF THE ARTICLE

Why-questions are complex and involve variability in their answers depending upon the need of the user. There is a need to accurately understand the users' requirement from question in order to retrieve appropriate documents and answer candidates. The users' need from the question is highly dependent on determining the main question focus which sometimes requires reformulation in case of short redundant questions. Since finding one correct answer to non-factoid why-type questions require extensive analysis of questions which sometimes demands question reformulation which can be interactive or non-interactive. The paper contributes in designing an algorithm for reformulating question into query based on different question types proposed in the research by Breja and Jain (2017; 2018). This objective is achieved by carrying out following three steps:

1. Analyzing different components of question with their impact on reformulation.
2. Designing algorithm for reformulating question into query based on different why-type question.
3. Developing a user interface for why-question to query reformulation which outputs features of question, reformulates it and takes user feedback on the retrieved web pages.

DIFFERENT COMPONENTS OF QUESTION AND THEIR ROLE ON REFORMULATION

There are different characteristics of question which play crucial role in understanding the process of reformulation.

1. *Named Entity Recognition:* It is one of the subtasks for information extraction from natural language text. It identifies real world entities from the text. In python, there are two ways to identify named entities in a question (1) using Stanford Core NER and NLTK which recognizes three classes of named entities viz. 'Location', 'Person', 'Organization' and 'O' as a background tag which don't fit any of the three labels. (2) using SPACY which supports various types of entities such as 'Person', 'NORP', 'FAC', 'ORG', GPE, 'LOC', 'PRODUCT', EVENT, DATE, TIME, PERCENT, MONEY and many more (Levengood, 2020). The paper utilizes SPACY to identify named entities from the question that reflects the role of each entity in a question.
2. *Tokenization:* Tokenization is a process of breaking text into smallest unit called tokens. Each why-question is tokenized and separated into tokens using 'word_tokenize' in NLTK.
3. *POS Tagging:* POS tagging is a process to assign part of speech tag to each tokenized word in a sentence. The paper applies POS tagging using NLTK to assign tags to each token of why-type question.

4. *Lemmatization:* Lemmatization is a process of removing inflectional endings of words and output their base or dictionary form, termed as lemma. It is better than stemming as it considers the context of each word to perform morphological analysis on them. In Python, there are 9 approaches to implement lemmatization; (1) WordNet, (2) WordNet + POS tag, (3) TextBlob, (4) TextBlob +POS tag, (5) spaCy, (6) TreeTagger, (7) Pattern, (8) Gensim, and (9) Stanford CoreNLP (Prabhakaran, 2021) . The paper applies spaCy module to lemmatize each word of why-question as it overcomes limitations from other approaches.
5. *Sentiment analysis:* Sentiment analysis also termed as opinion mining is a process of determining polarity of text, whether it is a document, paragraph, sentence or phrase. Polarity is categorized as positive, negative or neutral which ultimately reflects the opinion, attitude or emotions of speaker/writer. The paper performs sentiment analysis of question using VADER tool. It is an efficient to predict the positivity or negativity of a text with their magnitude.
6. *Noun phrase extraction:* This helps to determine a list of noun phrases in the question text. It is utilized using noun_phrases property of TextBlob in python.
7. *Constituency parsing:* Constituency parsing utilizes constituent-based grammar to analyze and extract the constituents of a text which represents its internal structure. It breaks the sentences into its constituents according to phrase structure rules of grammar. These rules help to determine the ordering and hierarchical structure of constituents in sentence. Each user input question is parsed to analyze the syntactic structure of its constituents such as NP (noun phrases), VP (verb phrases) and PP (Prepositional phrases) and many more. It is implemented using StanfordCoreNLP parse method (Bengfort, 2018).
8. *Dependency parsing with tree formation:* Dependency parsing is another type of parsing which analyzes the grammatical structure of text by considering the dependencies involved between each words in a sentence. It uses dependency-based grammars to analyze the syntactic and semantic dependencies with relationships between tokens of a sentence. Dependency relationship of each user question is constructed using displacy method of spacy and the whole dependency tree is visualized by converting spacy tree to nltk tree (Bengfort, 2018).

ALGORITHM TO REFORMULATE EACH QUESTION TYPE

This section discusses an algorithm which is proposed to reformulate Why-type questions into an appropriate query. The reformulated query helps in better retrieval of documents if posed on search engines.

There are positive and negative why-type questions. Negativity affects the query reformulation. A taxonomy is proposed for why-type questions by Breja and Jain (2018) which is categorized as (1) Informational which seeks reasoning about the facts, (2) Historical which seeks reason of the events occurred in past, (3) ComparativeSituational which asks reasoning for comparison of events occurred at a particular situation or circumstance, (4) Situational seeks reasoning for events occurred at a particular situation and (5) Opinionated which asks about the opinions on some person or product.

Rules are analysed for each type of why-questions and important focus terms are visualized which play a significant role to formulate query posed on search engine.

```
Algorithm 1. Refomulation of Why-Question to query based on their
grammatical structure
Input: User Input Why-Question
Output: Reformulated query
Grammar Representation taken in algorithm: * represents zero or
more occurrences and + represents one or many occurrences.
Notations used:
NN (singular noun), NNS (plural noun), NPh (NounPhrase), IN
(preposition/subordinating conjunction), PRP (personal pronoun),
```

VBN (verb past principle), VBD (verb past tense), VB (verb), VBPh (verb phrase), VBP (verb, present tense not 3rd person singular), VBZ(verb, present tense with 3rd person singular), VBG (verb gerund), RB (adverb), RBR (comparative adverb), MD (modal), TO (infinite marker (to)), JJ (adjective), JJR (comparative adjective), CD (cardinal digit), WRB (wh-adverb), DT (determiner), CC (coordinating conjunction), | (or)

NPh -> NPh NN | NN NPh | NPh IN NN | PRP (NN)* | PRP NNS | NPh

VBPh -> VBN IN | RB VBN | RB VBD | MD VB | RB VBN

Procedure:
Step1: Find POS tagging of the question using NLTK
Step 2: Apply classification algorithm on the questions to find the type of questions
Step 3: If the type of why-question is Informational type:
Step 3a: Check if the pattern of question is Why VBP|VBZ NPh|NN (TO (VBPh)* (JJ)* (IN)* (JJ)* NPh)$^+$, the reformulated query is {NPh VBP|VB NPh|NN because} or {NPh (IN JJ | JJ IN) (NPh)* because } (if no VBP|VB present in case of VBZ)
Step 3b: Check if the pattern of question is Why NPh (VBZ|VBG|VBN|VBPh|VBP)$^+$ (IN NN|NPh|JJ|CD))*, the reformulated query is {(NPh)$^+$ (VNP|VBPh)* (NPh)* IN NPh (IN CD)* because}
Step 4: If the type of why-question is Historical type:
Step 4a: The pattern of question is Why VBD NPh (RB)* VBP (NPh)* (CD|WRB|TO|IN)* (VB)* (RB | IN)* (CD|NPh)* and the reformulated query is {NPh (RB)* VBP (past tense) NPh (RB)* (IN)* (CD)* (NPh)* because}
Step 5: If the type of why-question is Situational type:
Step 5a: If the pattern of question is Why (VBZ|VBP)* (NPh|NNS|NN) (JJ TO|RBR)* (VBZ|VBN|VBP)* (IN NPh | NNP)$^+$ and VBP comprises one of RB VBN | RB VBD, the reformulated query is { NPh (JJ|TO|RBR)* (VBN|VBP|VBD) (IN NPh)$^+$ because} and requires user input regarding significance of RB.
Step 5b: If the pattern of question is Why NPh VBP TO NPh IN NPh IN CD, the reformulated query is { NPh VBP NPh IN NPh IN CD because}
Step 5c: If the pattern of question is Why VBP VB VBP IN VBP, the reformulated query is { VBP VB VBP IN VBP because}

```
Step 6: If the type of why-question is ComparativeSituational:
Step 6a: If the pattern of question is WHY VBP NPh (VBP (NPh)* (CC|IN) (NPh)* VBP
                (JJ)*), the reformulated query is {NPh (VBP (NPh)* (CC|IN) (NPh)* VBP
                (JJ)*) because} and if NPh is others, perform anaphora resolution to find related
                entity to it.
Step 6b: If the pattern of question is WHY VBZ (DT)* (NN|NPh) (VBD|RBR)* (CC|IN)
                (NPh)+ (CC RB JJ IN NPh)*, the reformulated query is {NN|NPh) (VBD|RBR)*
                (CC|IN) (NPh)+ (CC RB JJ IN NPh)* because}
Step 6c: If the pattern of question is WHY VBP (NN|NPh) CC (NN|NPh) (NNS)* VBP
                ((RB)* JJ), the reformulated query is {(NN|NPh) CC (NN|NPh|NNS)+ (RB)* JJ

                 because}
Step 6d: If the pattern of question is WHY (NPh VBP TO VBP)* NPh (VBZ)*
                 (JJR|RBR|JJ)* IN (JJ|NPh|NN|NNS), the reformulated query is {(VBP TO VBP)*
                 NPh (JJR|RBR|JJ)* IN (JJ|NPh|NN|NNS) because} and if the question begins
                 with (VBP TO VBP)* or ending with JJ, requires user input to find the actual
                 need.
Step 7: If the type of why-question is Opinionated type:
Step 7a: If the pattern of question is Why NPh (VBD|VBZ) (JJ| NPh) IN NPh, the
                 reformulated query is {NPh (VBD|VBZ) (JJ| NPh) IN NPh because}
Step 7b: If the pattern of question is Why (VBZ|(VBD RB)) (DT)* NPh VBP (NPh)* IN NPh
                (VBD)* (WRB NPh)* VBP (NPh)* (JJR IN DT NPh)*, the reformulated query is
               {NPh (VBD|VB)* (NPh)* IN NPh (VBD|VBP)* because}
Step 7c: If the pattern of question is Why (VBP|VBD) NPh (RB VBP)* (VBPh | VB)* (NPh |
                (TO VB JJ)* IN (NPh|VBPh) | (RP VBN))*, the reformulated query is {NPh (VBPh|
                 VBP|VB)+ (NPh|IN|RP|JJR|TO)*(NPh|VBPh)* because}
```

The example of above algorithm is illustrated from Table 1 to Table 5.

IMPLEMENTATION DETAILS

This section describes implementation for user interface designed for question reformulation. The user interface provides an ease to user with functionalities of question. The implementation is performed using Tkinter module of Python and functionalities are performed using NLTK module of Python.

Table 1. Example of Informational Why-Question and their query corresponding to their different patterns

Informational Why-type Questions			
Pattern 1: Why VBP|VBZ NPh|NN (TO (VBPh)* (JJ)* (IN)* (JJ)* NPh)+			
Question Example	**Focus words**	**Focus terms**	**Query**
Why are hush puppies called hush puppies?	Hush puppies called hush puppies	NPh VBPh NPh	NPh VBP VBPh NPh because
Why does sugar taste sweet?	Sugar taste sweet or sugar sweet	NPh VB NN	NPh VB (if VBZ) NN because
Why is cleanliness an important requirement for contact lenses?	Cleanliness an important requirement for contact lenses	NN NPh NPh IN NPh	In case of VBZ, if no VB, then add IN NN NPh NPh IN NPh because
Why is “fish” referred to as “brain food”?	fish referred brain food	NPh VBPh NPh	NPh VBPh NPh because
Pattern 2: Why NPh (VBZ|VBG|VBN|VBPh|VBP)+ (IN NN|NPh|JJ|CD))*:			
Why will cereal farmers rejoice?	Cereal farmers rejoice	NPh VBPh	NPh VBPh because
Why groups are commutative?	Groups are commutative	NPh VBP JJ	NPh VBP JJ because
Why mathematics is foundation for computer science?	Mathematics foundation for computer science	NPh NPh IN NPh	NPh NPh IN NPh because
Why Lionel Messi known as God of Football?	Lionel Messi known God of Football	NPh VBPh NPh IN NPh	NPh VBPh NPh IN NPh because

Table 2. Example of Historical Why-Question and their query corresponding to their pattern

Historical Why-Questions			
Pattern 1: Why VBD NPh (RB)* VBP (NPh)* (CD|WRB|TO|IN)* (VB)* (RB | IN)* (CD|NPh)*			
Why did the chicken cross the road?	The chicken crossed the road	NPh VBP (past tense) NPh	NPh VBP (past tense) NPh because
Why was Pearl Harbor bombed?	Pearl Harbor bombed	NPh VBP (past tense)	NPh VBP (past tense) because
Why United States entered World War 2?	United States entered World War 2	NPh VBP (past tense) NPh	NPh VBP (past tense) NPh because
Why was Paris given board boulevards after 1848?	Paris given board boulevards after 1848	NPh VBP NPh IN CD	If CD- add In conjunction- plays significance NPh VBP NPh **IN CD** because
Why did india not win the women’s cricket t20 world cup in 2020?	India not win the women’s cricket t20 world cup in 2020	NPh (RB VBP) NPh IN CD	If CD- add In conjunction- plays significance NPh **(RB VBP)** NPh **IN CD** because

Table 3. Example of Situational Why-Question and their query corresponding to their patterns

Situational Why-type Questions			
Pattern 1: Why (VBZ|VBP)* (NPh|NNS|NN) (JJ TO|RBR)* (VBZ|VBN|VBP)* (IN NPh | NNP)+			
Why India is still come under the category of developing country?	India come under the category of developing country	NPh VBN IN NPh IN NPh	NPh VBN IN NPh IN NPh because
Why is street food risky to eat in india during monsoon?	Street food risky to eat in India during monsoon	NPh JJ TO VBP IN NPh IN NPh	NPh JJ TO VBP IN NPh IN NPh because
Why does temperature change during the seasons?	Temperature change during the seasons	NN VBP IN NPh	NPh VBN IN NPh because
Pattern 2: Why VBP VB VBP IN VBP			
Why should chicken be well cooked before eating?	Chicken be well cooked before eating	VBP VB VBP IN VBP	VBP VB VBP IN VBP because

Table 4. Example of ComparativeSituational Why-Question and their query corresponding to their patterns

ComparativeSituational Why Questions			
Pattern 1: WHY VBP NPh (VBP (NPh)* (CC|IN) (NPh)* VBP (JJ)*)			
Why do stock prices rise and fall?	Stock prices rise and fall	NPh VBP CC VBP	NPh VBP CC VBP because
Why do some chickens lay brown eggs while others lay white?	Some chickens lay brown eggs while others lay white	NPh VBP NPh IN NPh VBP JJ	NPh VBP NPh IN NPh VBP JJ because
Pattern 2: WHY VBZ (DT)* (NN|NPh) (VBD|RBR)* (CC|IN) (NPh)+ (CC RB JJ IN NPh)*			
Why is the sky red at sunset and also colorful at sunrise?	Sky red at sunset and colourful at sunrise	VBD IN NPh CC JJ IN NPh	VBD IN NPh CC JJ IN NPh because
Why does a seashell sound like the ocean?	a seashell sound like the ocean	NPh NN IN NPh	NPh NN IN NPh because
Why is mind and muscle memory important for exercise enthusiasts?	Mind and muscle memory important for exercise enthusiasts	NPh CC NPh JJ IN NPh	NPh CC NPh JJ IN NPh because
Why is obesity higher in developing and developed countries?	Obesity higher in developing and developed countries	NN RBR IN NPH	NN RBR IN NPH because
Pattern 3: WHY VBP (NN|NPh) CC (NN|NPh) (NNS)* VBP ((RB)* JJ)			
Why do movie and TV stars get paid so much?	Movie and TV stars get paid so much	NN CC NN NNS VBP RB JJ	NN CC NN NNS VBP RB JJ because
Why Zomato and Swiggy are getting famous?	Zomato and Swiggy getting famous	NPh CC NPh VBP JJ	NPh CC NPh VBP JJ because
Pattern 4: WHY (NPh VBP TO VBP)* NPh (VBZ)* (JJR|RBR|JJ)* IN (JJ|NPh|NN|NNS)			
Why light travels faster than sound?	Light travels faster than sound	NPh RBR IN NN	NPh RBR IN NN because
Why Instagram is popular than Facebook nowadays?	Instagram popular than Facebook nowadays	NPh JJ IN NPh NNS	NPh JJ IN NPh NNS because
Why tourists prefer to visit Niagara Falls than Venice Beach?	Tourists prefer to visit Niagara Falls than Venice Beach	NPh VBP TO VBP NPh IN NPh	NPh VBP TO VBP NPh IN NPh because

Table 5. Example of Opinionated Why-Question and their query corresponding to their patterns

<table>
<tr><th colspan="4">Opinionated Why-type Questions</th></tr>
<tr><td colspan="4"><b>Pattern 1: Why NPh (VBD|VBZ) (JJ| NPh) IN NPh</b></td></tr>
<tr><td>Why B.B. King named his guitar as “Lucille”?</td><td>B.B. King named his guitar as “Lucille”</td><td>NPh VBD NPh IN NPh</td><td>NPh VBD NPh IN NPh because</td></tr>
<tr><td>Why internet is important for your life?</td><td>Internet important for your life</td><td>NPh JJ IN NPh</td><td>NPh JJ IN NPh because</td></tr>
<tr><td colspan="4"><b>Pattern 2: Why (VBZ|(VBD RB)) (DT)* NPh VBP (NPh)* IN NPh (VBD)* (WRB NPh)* VBP (NPh)* (JJR IN DT NPh)*</b></td></tr>
<tr><td>Why didn’t Socrates leave Athens after he was convicted?</td><td>Socrates didn’t leave Athens after he was convicted</td><td>NPh VBD VB NPh IN NPh VBD VBP</td><td>NPh VBD VB NPh IN NPh VBD VBP because</td></tr>
<tr><td>Why is the gastrointestinal tract of animals that eat grass longer than a humans?</td><td>Gastrointestinal tract of animals that eat grass longer than a humans</td><td>(NPh) IN NPh WDT VBP NPh JJR IN NPh</td><td>(NPh) IN NPh WDT VBP NPh JJR IN NPh</td></tr>
</table>

The reformulation interface opens with a welcome page indicating ‘Why-Question to Query Reformulator’. Next page has an input label which asks user to input Why-question. The input question is processed in the next window to calculate its functionalities which are : Classification into its type and answer type, Named entity recognition, POS tagging, Tokenization, constituency parsing, dependency parsing with tree formation, Sentiment analysis, lemmatization, noun phrase extraction; the details of which are discussed in Section 4. The functionality button is clicked by the user and its result is displayed on the Python console.

Next window consists of the button for ‘question to query’ which reformulates input why-question to query according to an algorithm discussed in Section 5. The output query is posed on the Google search engine to extract relevant web pages to it for which user feedback is taken. The performance of user feedback is discussed in Section 7 below.

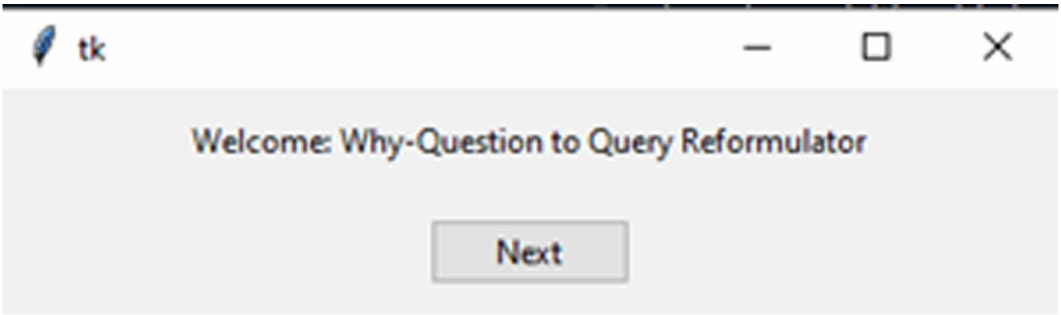
The snapshots of the user interface developed are illustrated below:

RESULTS AND DISCUSSION

A user interface is implemented which takes input why-question from user and applies various functionalities to compute the value of various components such as sub-classification and determining its type and answer type, named entities recognition, tokenization, POS tagging, constituency and dependency parsing, noun phrase extraction. After determining its type and functionalities, patterns of why-type questions are analyzed. For each pattern, rules are formulated which finds the important terms from question and helps to formulate query that can be posed on search engine to extract relevant web pages. The web pages returned are rated by the 10 users of different age groups on the scale of 1 to 10 which measures how many pages out of 10 returned pages are relevant to users.

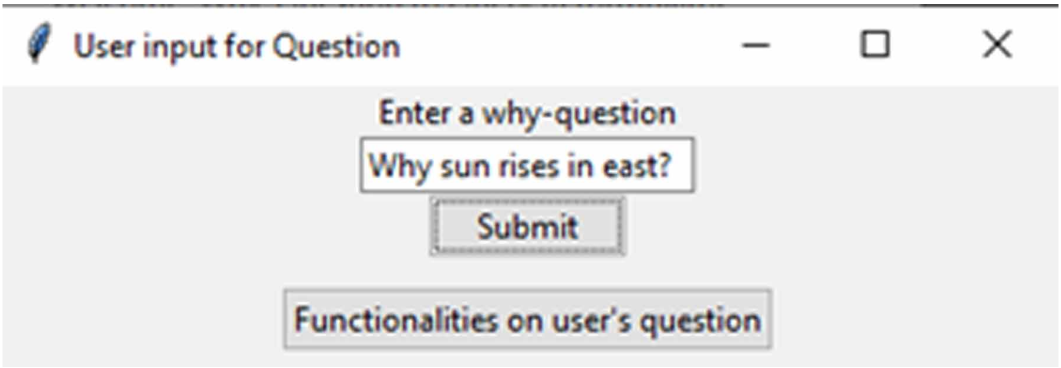
10 questions of each question type are reformulated and posed on Google search engine for a set of web pages. Returned web pages for each question are distributed to 10 users of different age group to take their feedback on the relevancy of the returned web pages. Further in order to calculate precision of an algorithm, a group technique is adopted where 10 users collaborate together as a team and arrive at a scoring of pages returned for each question type. This further helps to calculate the final precision which finds the ratio of total number of relevant documents retrieved by the total number of documents retrieved (“Precision and Recall”, n.d.)

Figure 1. Welcome page of user interface



Welcome: Why-Question to Query Reformulator

Figure 2. Asking user to input why-question and button to compute its functionalities



Why sun rises in east?

Figure 3. Buttons for calculating different components of question

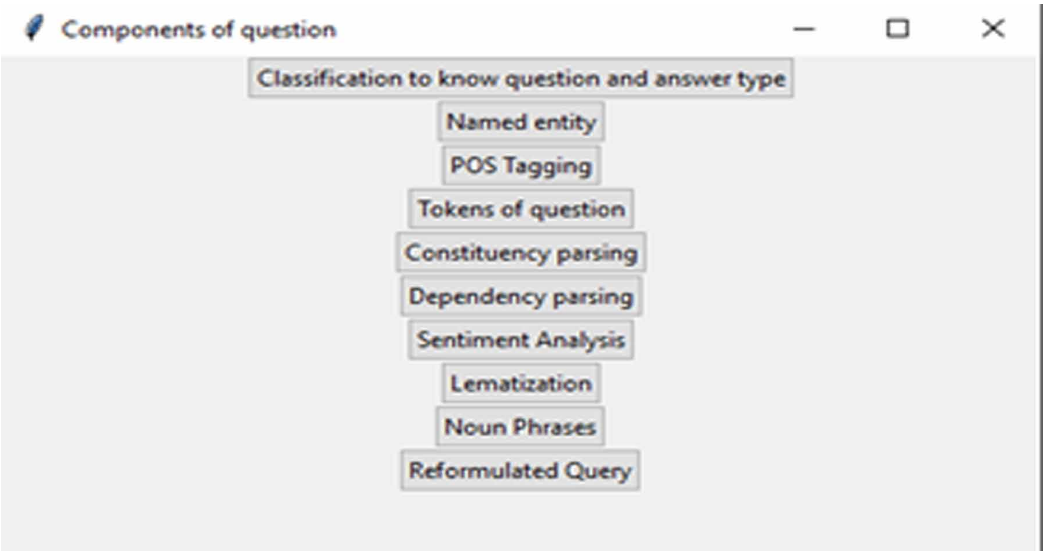


Figure 4. Buttons for finding focus, reformulated query, searching on google and taking user feedback on retrieved web pages

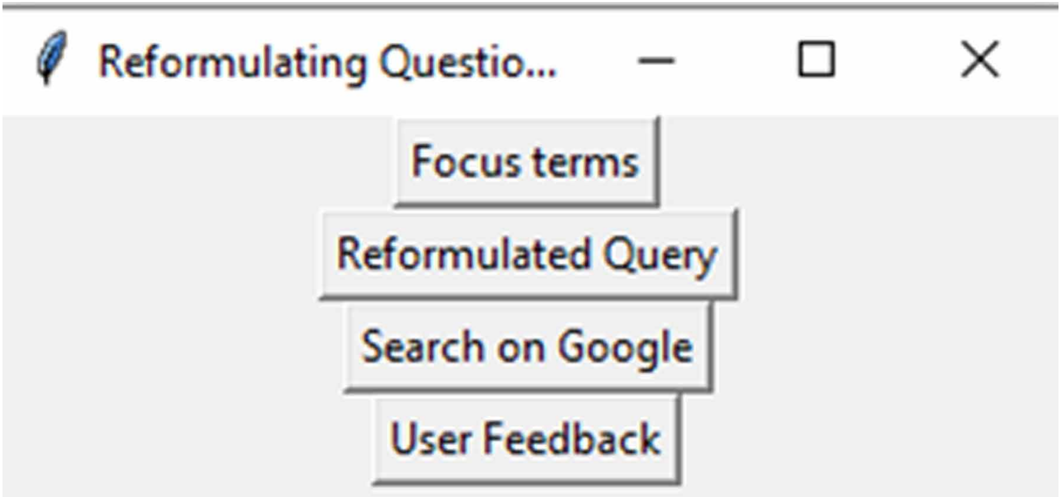


Figure 5. Scaler to take input from user regarding satisfaction of retrieved Google results based on reformulated query

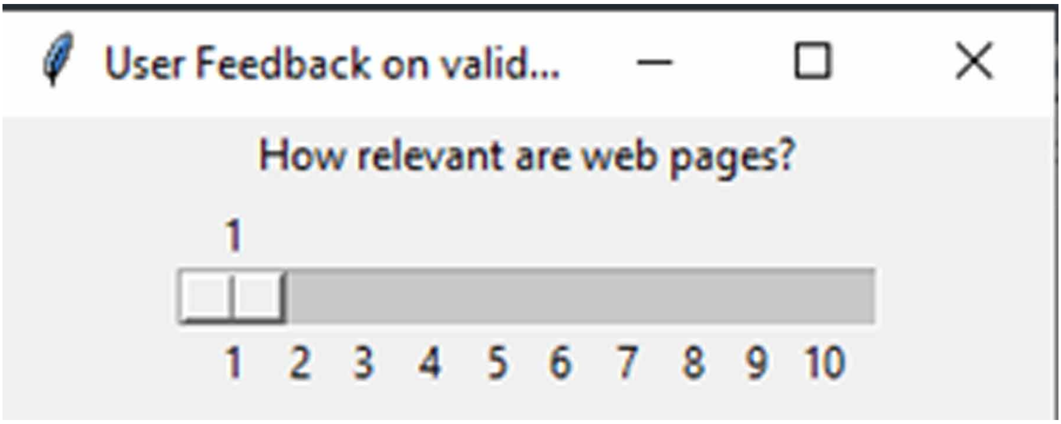
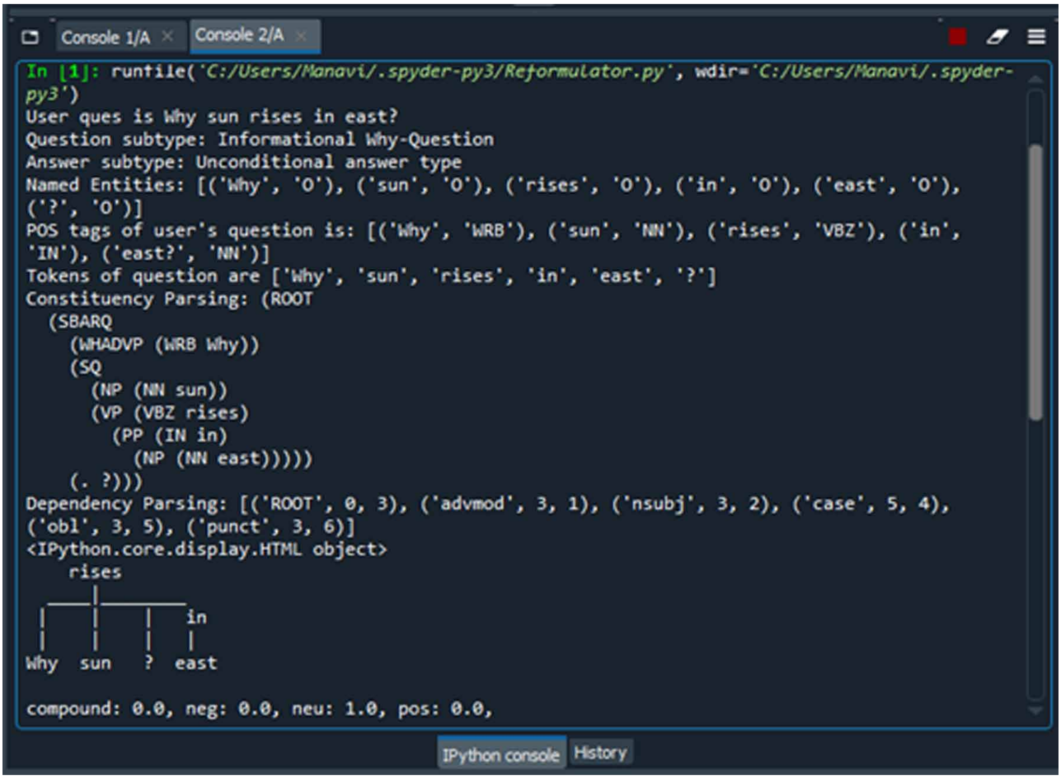
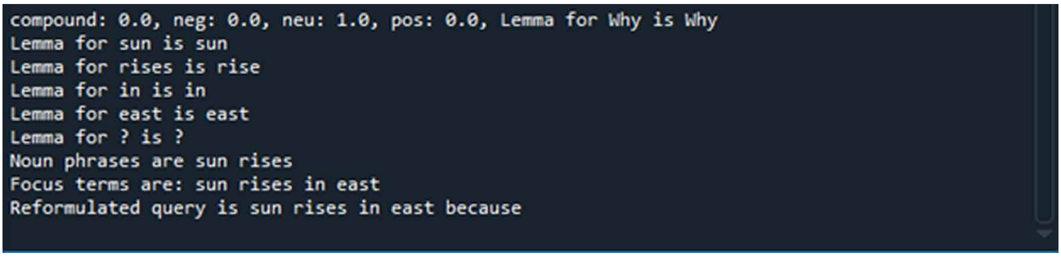


Figure 6. Output of different functionalities applied on sample user input question ‘Why sun rises in east?’



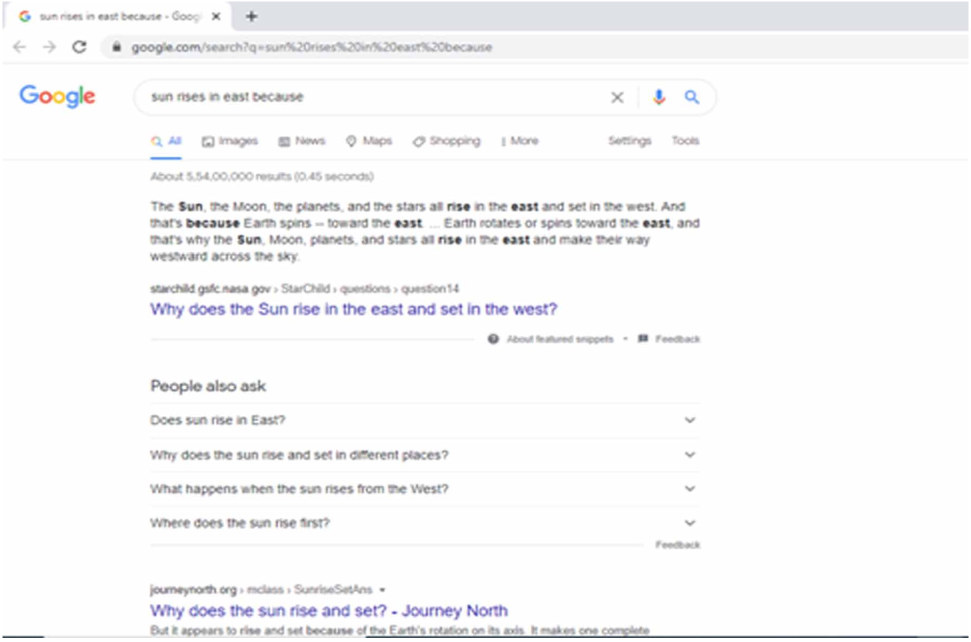
```
In [1]: runfile('C:/Users/Manavi/.spyder-py3/Reformulator.py', wdir='C:/Users/Manavi/.spyder-
py3')
User ques is Why sun rises in east?
Question subtype: Informational Why-Question
Answer subtype: Unconditional answer type
Named Entities: [('Why', 'O'), ('sun', 'O'), ('rises', 'O'), ('in', 'O'), ('east', 'O'),
('?', 'O')]
POS tags of user's question is: [('Why', 'WRB'), ('sun', 'NN'), ('rises', 'VBZ'), ('in',
'IN'), ('east?', 'NN')]
Tokens of question are ['Why', 'sun', 'rises', 'in', 'east', '?']
Constituency Parsing: (ROOT
  (SBARQ
    (WHADVP (WRB Why))
    (SQ
      (NP (NN sun))
      (VP (VBZ rises)
        (PP (IN in)
          (NP (NN east)))))
    (. ?)))
Dependency Parsing: [('ROOT', 0, 3), ('advmod', 3, 1), ('nsubj', 3, 2), ('case', 5, 4),
('obl', 3, 5), ('punct', 3, 6)]
<IPython.core.display.HTML object>
    rises
  ____|________
 |    |    |   in
 |    |    |   |
Why  sun   ?  east

compound: 0.0, neg: 0.0, neu: 1.0, pos: 0.0,
```

Figure 7. Output of lemmatization, focus, noun phrases and reformulated query to the input user question ‘Why sun rises in east?’



```
compound: 0.0, neg: 0.0, neu: 1.0, pos: 0.0, Lemma for Why is Why
Lemma for sun is sun
Lemma for rises is rise
Lemma for in is in
Lemma for east is east
Lemma for ? is ?
Noun phrases are sun rises
Focus terms are: sun rises in east
Reformulated query is sun rises in east because
```

Figure 8. Google web pages corresponding to the reformulated query ‘sun rises in east because’


sun rises in east because - Googl
google.com/search?q=sun%20rises%20in%20east%20because
Google
sun rises in east because
All
Images
News
Maps
Shopping
More
Settings
Tools
About featured snippets
Feedback
Feedback

About 5,54,00,000 results (0.45 seconds)

The **Sun**, the Moon, the planets, and the stars all **rise** in the **east** and set in the west. And that's **because** Earth spins -- toward the **east**. ... Earth rotates or spins toward the **east**, and that's why the **Sun**, Moon, planets, and stars all **rise** in the **east** and make their way westward across the sky.

starchild.gsfc.nasa.gov › StarChild › questions › question14
Why does the Sun rise in the east and set in the west?

People also ask

Does sun rise in East?

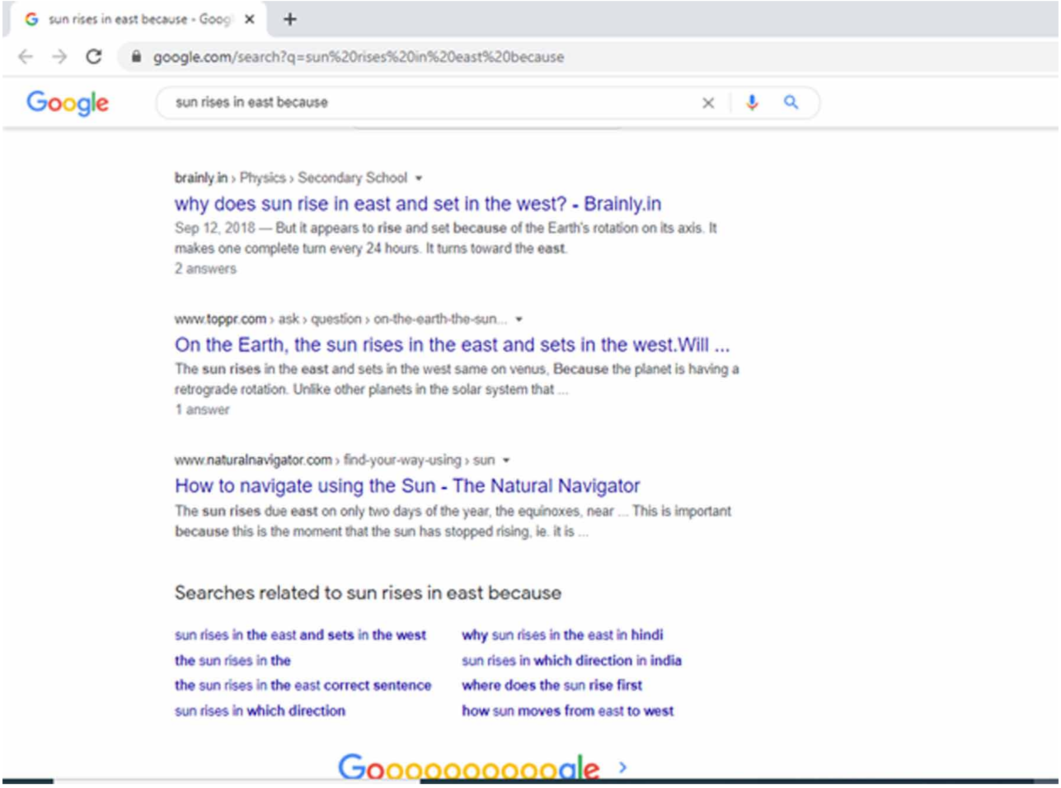
Why does the sun rise and set in different places?

What happens when the sun rises from the West?

Where does the sun rise first?

journeynorth.org › mclass › SunriseSetAns
Why does the sun rise and set? - Journey North
But it appears to rise and set because of the Earth's rotation on its axis. It makes one complete

Figure 9. Google web pages corresponding to the reformulated query ‘sun rises in east because’ contd.


sun rises in east because - Googl
google.com/search?q=sun%20rises%20in%20east%20because
Google
sun rises in east because
Gooooooooooogle

brainly.in › Physics › Secondary School
why does sun rise in east and set in the west? - Brainly.in
Sep 12, 2018 — But it appears to **rise** and set **because** of the Earth's rotation on its axis. It makes one complete turn every 24 hours. It turns toward the **east**.
2 answers

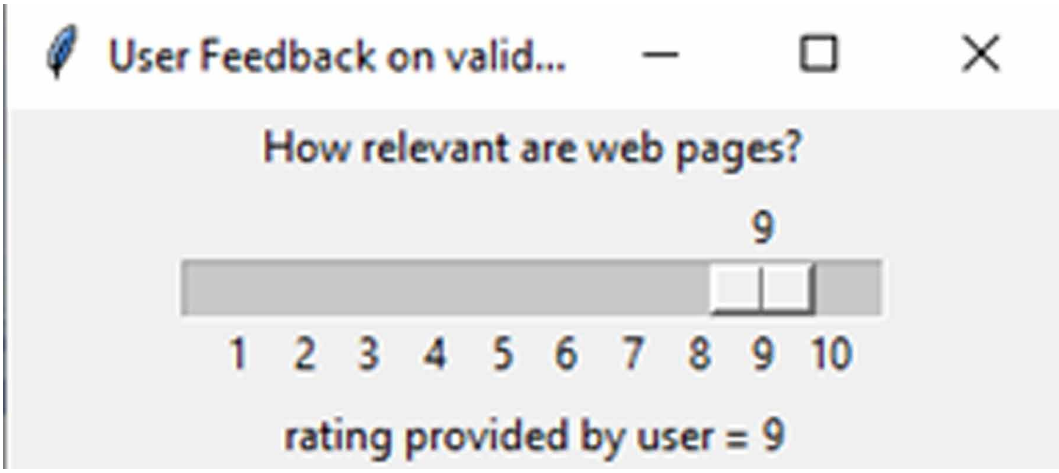
www.toppr.com › ask › question › on-the-earth-the-sun...
On the Earth, the sun rises in the east and sets in the west.Will ...
The **sun rises** in the **east** and sets in the west same on venus, Because the planet is having a retrograde rotation. Unlike other planets in the solar system that ...
1 answer

www.naturalnavigator.com › find-your-way-using › sun
How to navigate using the Sun - The Natural Navigator
The **sun rises** due **east** on only two days of the year, the equinoxes, near ... This is important because this is the moment that the sun has stopped rising, ie. it is ...

Searches related to sun rises in east because

sun rises in **the east and sets** in **the west**
the sun rises in **the**
the sun rises in **the** east **correct sentence**
sun rises in **which direction**
why sun rises in **the** east **in hindi**
sun rises in **which direction in india**
where does the sun **rise first**
how sun **moves from** east **to west**

Figure 10. Scaler to input user feedback on the satisfaction of retrieved web pages to the query



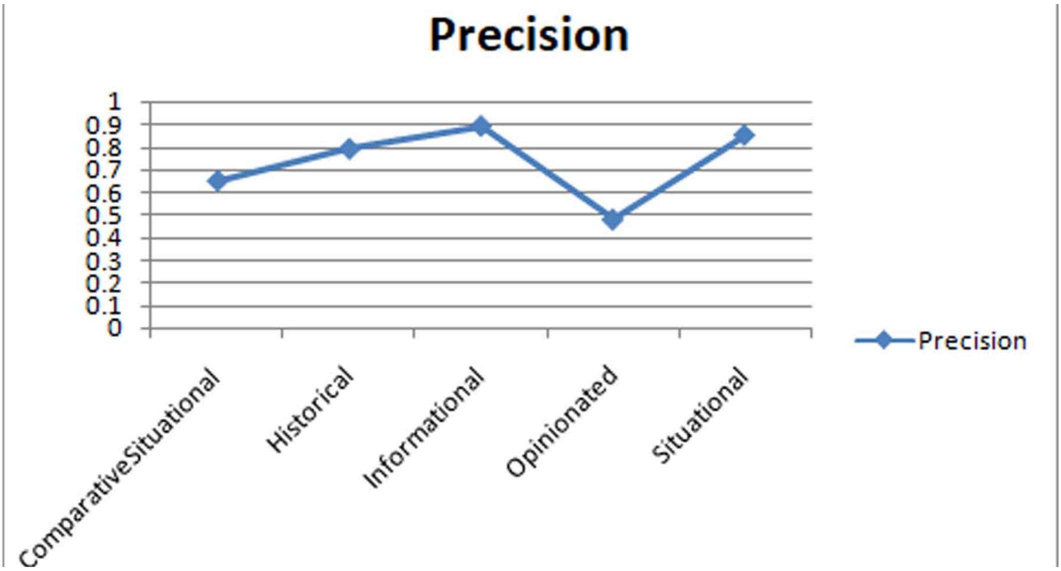
User Feedback on valid...

How relevant are web pages?

9

rating provided by user = 9

Figure 11. Precision of users feedback based on different Why-type questions



$$\text{Precision}= \frac{Total\,number\,of\,relevant\,documents\,retrieved\,by\,search}{Total\,number\,of\,documents\,retrieved\,by\,search} \quad (1)$$

Table 6, briefs the precision achieved for each Why-question type which is illustrated further in Figure 11.

The results of precision clearly depict that informational why-type questions are best answered by Google search. In both ComparativeSituational and situational questions, input is required for some type of questions which needs further clarification on the part of understanding the need of user. In Historical type questions, most of the questions can be appropriately answered but some questions

Table 6. Precision achieved for each Why-question type

QuestionType	Precision
ComparativeSituational	0.65
Historical	0.79
Informational	0.89
Opinionated	0.48
Situational	0.85

which seek answering with respect to the particular time period suffers with inappropriate answers. The case of opinionated why-question shows very drastic results because answering to such questions differ with person's opinion and thus can't be answered properly if directly posed on search engine. Thus, the results very well infer that reformulation alone can't improve the accuracy of why-question answering but interaction of user is also required in many questions which can clarify their actual demand in the answering.

COMPARISON OF OUR PROPOSED APPROACH WITH OTHER WORK ON QUESTION REFORMULATION

Table 7 discusses the work on question reformulation with our proposed approach of question reformulation on Why-type QAS.

Table 7. Comparison of our proposed approach with other work of question reformulation on QAS

References	Methodology for reformulation	Question type	Performance
Umamehaswari et al. (2012)	Semantic based technique, patterns based on lexical, syntactic and semantic constraints are generated	Who, what, where, when, how, which	0.498 precision with candidate answer, 0.588 with generated patterns
Iturbe Herrera et al. (2021)	Different grammatical elements identified on question classes to locate question focus, CNN was trained on dataset	TREC V2, WebQues dataset involving what, when, who, where question type	96.84% accuracy with TREC dataset, 90.91% on WebQues, 87.37% on WikiMovies, 82.5% on TREc10 and 96.63% on SimpleQues dataset
Esposito et al.(2021)	Query expansion approach comprising 4 steps viz. Question processing and expandable terms identification, candidate expansion terms extraction and contextualization, candidate expansion terms ranking and filtering and query formulation	Person, Entity, Location, Date, Description as top level categories, and Address, City, Region, Artifact as bottom level categories	Improvement in accuracy by 9.1% over the approaches based on MultiWordNet, to 15.7% over Word2vec model and 37.1% over an approach without QE.
Vakulenko et al. (2021)	Question Rewriting (QR) as a component of QA task. Different question rewriting (QR) models were proposed which boosts the performance of answer extraction	Conversational question answering	Achieved maximum 0.81 ROUGE score on Transformer ++ QR model in comparison to 0.84 human performance on CANARD test set with 0.90 ROUGE score on TREC CAsT test set in comparison to 1.00 human performance.
Verberne (2010)	No reformulation concept was introduced	Why-type question answering	Success@150 is 78.5%, for 21.5% of questions, there was no answer retrieved in top-150 documents retrieved
Proposed Work	Categorized Why-type questions, identified different patterns corresponding to each answer type, reformulates them into a query based on algorithm	Why-type question answering	Maximum precision of 0.89 for Informational Why-type questions

CONCLUSION

The paper performs reformulation of why-type questions to query depending on different classes of why-type questions according to taxonomy proposed by the research in Breja and Jain (2018). With an algorithm, a user interface is also designed which asks user to input why-question, calculates different components of question, reformulates it into a query, and finally takes feedback on the web pages retrieved by a reformulated query. The method achieves precision of 89% in Informational-type and 48% in Opinionated-type why-questions. In future, performance of opinionated questions will be improved by incorporating interactive query refinement techniques where system will interact with user and take input at subsequent steps which will help the system to understand properly the need of such questions.

REFERENCES

Bengfort, B. (2018, June 22). *Syntax Parsing with CoreNLP and NLTK*. bbengfort.github.io

Bernhard, D. (2010, August). Query expansion based on pseudo relevance feedback from definition clusters. In *Coling 2010* (pp. 54–62). Posters.

Boldi, P., Bonchi, F., Castillo, C., & Vigna, S. (2011). Query reformulation mining: Models, patterns, and applications. *Information Retrieval*, *14*(3), 257–289. doi:10.1007/s10791-010-9155-3

Breja, M., & Jain, S. K. (2017). Why-type Question Classification in Question Answering System. In *FIRE* (pp. 149–153). Working Notes.

Breja, M., & Jain, S. K. (2018, September). Analysis of Why-Type Questions for the Question Answering System. In *European Conference on Advances in Databases and Information Systems* (pp. 265-273). Springer. doi:10.1007/978-3-030-00063-9_25

Esposito, M., Damiano, E., Minutolo, A., De Pietro, G., & Fujita, H. (2020). Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering. *Information Sciences*, *514*, 88–105. doi:10.1016/j.ins.2019.12.002

Habernal, I., Konopík, M., & Rohlík, O. (2012). Question Answering. In Next Generation Search Engines: Advanced Models for Information Retrieval (pp. 304-343). IGI Global. doi:10.4018/978-1-4666-0330-1.ch014

Herdagdelen, A., Ciaramita, M., Mahler, D., Holmqvist, M., Hall, K., Riezler, S., & Alfonseca, E. (2010, July). Generalized syntactic and semantic models of query reformulation. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (pp. 283-290). doi:10.1145/1835449.1835498

Iturbe Herrera, A., Castro Sánchez, N. A., & Mújica Vargas, D. (2021). Rule-Based Spanish Multiple Question Reformulation and their Classification using a Convolutional Neuronal Network. *Computación y Sistemas*, *25*(1). Advance online publication. doi:10.13053/cys-25-1-3895

Kangavari, M. R., Ghandchi, S., & Golpour, M. (2008). Information retrieval: Improving question answering systems by query reformulation and answer validation. *World Academy of Science, Engineering and Technology*, *48*, 303–310.

Kosseim, L., & Yousefi, J. (2008). Improving the performance of question answering with semantically equivalent answer patterns. *Data & Knowledge Engineering*, *66*(1), 63–67. doi:10.1016/j.datak.2007.07.010

Levengood, C. (2020, January 6). *Named Entity Recognition in Python with Stanford-NER and Spacy*. https://lvngd.com/blog/named-entity-recognition-in-python-with-stanford-ner-and-spacy/

Liu, Y. H., & Belkin, N. J. (2008, October). Query reformulation, search performance, and term suggestion devices in question-answering tasks. In *Proceedings of the second international symposium on Information interaction in context* (pp. 21-26). doi:10.1145/1414694.1414702

Moldovan, D., Pasca, M., Harabagiu, S., & Surdenau, M. (2003). Performance Issues and Error Analysis in an Open-Domain Question Answering System. *ACM Transactions on Information Systems*, *21*(2), 133–154. doi:10.1145/763693.763694

Molla, D. (2009). *From Minimal Logical Forms for Answer Extraction to Logical Graphs for Question Answering. Searching Answers: Festschrift in Honour of Michael Hess on the Occasion of His 60th Birthday*. MV-Wissenschaft.

Musa, R., Wang, X., Fokoue, A., Mattei, N., Chang, M., Kapanipathi, P., & Witbrock, M. (2018, November). Answering science exam questions using query reformulation with background knowledge. In *Automated Knowledge Base Construction*. AKBC.

Ooi, J., Ma, X., Qin, H., & Liew, S. C. (2015, August). A survey of query expansion, query suggestion and query refinement techniques. In *2015 4th International Conference on Software Engineering and Computer Systems* (pp. 112-117). IEEE. doi:10.1109/ICSECS.2015.7333094

Pires, R. M. P. (2012). *Query classification and expansion in just. Ask question answering system* [Doctoral dissertation]. Instituto Superior Tcnico.

Prabhakaran S. (n.d.). *Lemmatization Approaches with Examples in Python.* https://www.machinelearningplus.com/nlp/lemmatization-examples-python/

Precision and recall. (n.d.). Retrieved July 09, 2021, from https://en.wikipedia.org/wiki/Precision_and_recall

Soricut, R., & Brill, E. (2006). Automatic question answering using the web: Beyond the Factoid. *Information Retrieval*, *9*(2), 191–206. doi:10.1007/s10791-006-7149-y

Umamehaswari, M., Ramprasath, M., & Hariharan, S. (2012, December). Improved question answering system by semantic refomulation. In *2012 Fourth International Conference on Advanced Computing (ICoAC)* (pp. 1-4). IEEE. doi:10.1109/ICoAC.2012.6416824

Vakulenko, S., Longpre, S., Tu, Z., & Anantha, R. (2021, March). Question rewriting for conversational question answering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (pp. 355-363). doi:10.1145/3437963.3441748

Verberne, S. (2010). Search of the Why: Developing a system for answering why-questions. Academic Press.

Manvi Breja is a PhD Scholar from National Institute of Technology, Kurukshetra. She has completed her M.Tech from YMCA University, Faridabad, India. Her area of interest includes Information Retrieval, Data Mining and Natural Language Processing.

Sanjay Kumar Jain, PhD (MNNIT, Allahabad, India), is a Professor in the Department of Computer Engineering at NIT Kurukshetra, India. He is involved in research and has 30 year experience of teaching. His current research areas include database, data mining, information retrieval, and requirements engineering.