Journal of Cases on Information Technology
Volume 24 • Issue 3

Analyzing Linguistic Features for Answer Re-Ranking of Why-Questions

Manvi Breja, National Institute of Technology, Kurukshetra, India

Sanjay Kumar Jain, National Institute of Technology, Kurukshetra, India

https://orcid.org/0000-0003-1999-5530

ABSTRACT

Why-type non-factoid questions are ambiguous and involve variations in their answers. A challenge in returning one appropriate answer to users requires the process of appropriate answer extraction, re-ranking, and validation. There are cases where the need is to understand the meaning and context of a document rather than finding exact words involved in question. The paper addresses this problem by exploring lexico-syntactic, semantic, and contextual query-dependent features, some of which are based on deep learning frameworks to depict the probability of answer candidate being relevant for the question. The features are weighted by the score returned by ensemble ExtraTreesClassifier according to features importance. An answer re-ranker model is implemented that finds the highest ranked answer comprising largest value of feature similarity between question-and-answer candidate and thus achieving 0.64 mean reciprocal rank (MRR). Further, the answer is validated by matching the answer type of answer candidate and returning the highest-ranked answer candidate with matched answer type to a user.

KEYWORDS

Answer Candidate Extraction, Answer Re-Ranking, Contextual, ExtraTreesClassifier, Feature Importance, Lexical-Syntactic, Natural Language Processing, Semantic

INTRODUCTION

The advent of IBM's Watson (IBM Watson, 2020) has shown remarkable results in answering open-domain questions. Watson is now no longer treated as only a question answering (QA) system rather it also has ability to sense. Research in question answering domain has achieved high accuracy around 85% in answering factoid-type questions. However, today researchers are motivated to go beyond factoid QA, addressing non-factoid question answering such as 'why' and 'how' type questions. Some of the work from Verberne et al. (2010), Jansen and Surdeanu (2014), Fried and Jansen (2015), Oh et al. (2012, 2013) has been successful in answering open-domain non-factoid questions whereas Tran and Niederee (2018) has investigated deep learning frameworks for answering insurance and financial domain non-factoid questions but still performance is lower than factoid QAS such as IBM Watson.

The question answering system presents an accurate answer satisfying the need of user. Answering why-type questions is complex and the need is to tackle the complexity because of ambiguity and redundancy involved. The paper is contributed towards extracting answer candidates to a question by finding cue phrases reflecting cause-effect relations between terms in retrieved passages. Further

DOI: 10.4018/JCIT.20220701.oa10

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

an answer re-ranker is developed exploring the set of features based on similarity between question and answer candidates, weighted by feature importance scores. The method is able to achieve 0.64 Mean Reciprocal Rank (MRR) which significantly improves over other previous research works in why-type answer re-ranker.

The remaining paper is structured in the sections where Section 2 explores the previous work in answer re-ranking for why-type questions. Section 3 provides main focus of article describing issues and their solutions. Section 4 describes the system architecture utilized for research. Section 5 highlights the data which is setup for answer re-ranking. Section 6 presents features employed for re-ranking answer candidates and their relevance. Section 7 puts light on the algorithm proposed to weigh each feature set based on the importance of each feature. Section 8 briefs the algorithm used for answer validation process. Section 9 highlights the implementation details in Python. Section 10 compares our proposed work with other previous research works. Finally Section 11 concludes the work with future research directions.

BACKGROUND

A considerable work has already been done in addressing non-factoid type questions and improving answer re-ranker module. This section discusses major contributions in answer re-ranking of English and Japanese non-factoid questions.

Referring to English non-factoid QAS, Verberne et al. (2010) applied various machine learning techniques for ranking answer candidates. The authors explored linguistic features comprising tf-idf, syntactic overlaps, WordNet synsets, cue terms, common words in question & document title and WordNet relatedness. Learning to rank approaches categorized as pairwise, pointwise and listwise (Liu, 2011) were applied with their default hyperparameter settings where Support Vector Regression with its hyperparameter tuning performs best with MRR 0.350. Although the authors provided a good baseline for re-ranking answers but the need is to explore more semantic and contextual features with assigning a weight to each feature. Surdeanu et al. (2011) exploited non-factoid QA pairs from social QA sites, further trained answer re-ranker model by extracting various features comprising similarity (BM25), translation (IBM's Model 1), density/frequency, and web correlation (query-log correlation using PMI and chi-square) features using Perceptron and SVM-rank model to achieve 0.6416 & 0.6381 MRR respectively. Since QA pairs were retrieved from social QA sites, the authors could have considered more features such as number of votes, genres of QA pairs, user comments and answers rating that will help instigate ranking answers. Jansen et al. (2014) integrated lexical semantics with shallow and deep discourse features. The model was trained on open-domain Yahoo! Answers corpus comprising how-type QA pairs and Biology Textbook corpus comprising both how and why-type questions using SVM Rank thus achieved 26.57 P@1 and 49.31 MRR. Molino and Aiello (2014), Fried et al. (2015) have trained answer re-ranker module with lexical-semantic models on dataset of non-factoid how-type questions to achieve 0.7909 MRR and 0.5396 MRR respectively. The authors have significantly addressed the issue of answer re-ranking by learning word representations and finding direct & indirect associations between QA pairs. Tran and Niederee (2018) utilized deep learning frameworks for re-ranking answers of non-factoid questions from insurance & financial domain, achieving 0.616 MRR using SRanker mlp, 0.606 MRR using SRanker bilinear and 0.653 MRR using CARanker.

Considering the research in Japanese non-factoid QAS, Higashinaka and Isozaki (2008) represented answer candidates by causal expressions, content similarity between Q&A and causal relations representing cause & effect. These feature sets are utilized to train Answer Re-Ranker with RankingBoost and SVM rank thus achieving top 5 MRR as 0.305. The authors only considered 'cause' relation representing causality, other relations such as 'purpose', 'condition' need to be further explored. Oh et al. (2012) trained and tested answer re-ranker using TinySVM with features combining morphological and syntactic analysis, semantic word classes based on n-grams and sentiment analysis finding word & phrase polarity, thus achieving 0.336 P@1 and 0.377 MAP. Oh et al. (2013) identified intra and inter-sentential causal relations between terms and phrases through term matching, partial tree matching and excitation polarity matching (a concept introduced by Hashimoto et al. (2012)) for determining the best answer candidates. The approach was trained by TinySVM with linear kernel achieving 0.418 P@1 and 0.41 MAP. The authors have built a foundation for answer re-ranking which can be utilized for other lingual why-QAS. Sakamoto (2015) combined enhanced HITS algorithm with graph-based model on Japanese WebQAS. The authors adopted the concept used in TextRank by calculating the hub and authority scores to represent answer candidates achieving 0.575 MRR for top 10 results. This methodology can also be utilized for multi-document summarization.

MAIN FOCUS OF THE ARTICLE

This section brings into the issues faced in why-question answering with their solutions:

Issue 1: While addressing why-type questions, there is a lot of variability involved owing to the context and requirement of user, thus presenting the challenge of returning one accurate answer to a user.

Solution: The paper has addressed this issue by re-ranking answer candidates based on the scoring values of three types of features i.e. (1) lexico-syntactic (2) semantic and (3) contextual. The similarities between questions and their answer candidates are based on query-dependent features which measures the relevancy of answer candidate to a question.

Issue 2: Understanding the parameters which are crucial for answering why-type questions in order to match the performance with factoid question-answering.

Solution: The paper has identified features relevant to NLP concepts. The importance of features is determined by calculating scores using ensemble feature selection technique 'ExtraTreesClassifier'.

Issue 3: Understanding the intent of users from the question and answering them accordingly.

Solution: The paper has utilized the concept of answer type matching for returning one appropriate answer to a question during answer validation phase. The question and answer types are determined according to a taxonomy proposed by Breja and Jain (2018) which further helps in scoring answer candidates.

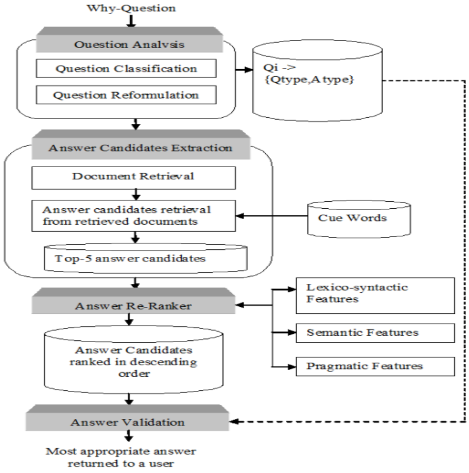
Thus the article proposes an answer re-ranking and answer validation model. Training for re-ranking answers is performed on the integrated feature-sets comprising lexico-syntactic, semantic and contextual features as detailed in Section 6. Lexico-syntactic features find relatedness between question and their respective answer candidates based on the terms and pos-tags involved in them. Semantic features measure the similarity by finding the synsets of words present in question and answer candidates. Contextual features capture the context of question and answer sentences using different sentence embedding techniques. The importance of each type of feature is computed by a proposed algorithm based on MRR value. Further the importance of feature in each set is determined by measuring their usefulness for answering why-type questions by determining their scores using ensemble technique. Answer validation is performed by matching the answer type of highest ranked relevant answer and determining the performance by finding percentage of questions having matched answer type at rank 1, rank 2, rank 3 and 4.

SYSTEM ARCHITECTURE

This section explains the proposed why-type question answering system involving four modules (1) Question Analysis (2) Answer candidates extraction (3) Answer Re-Ranker and (4) Answer Validation.

Figure 1 illustrates the systemized architecture of why-QAS.

Figure 1. System architecture for the proposed why-type question answering system



Question Analysis: Question Analysis incorporates two sub-modules (1) Question classification parses the user's input why-question and determines question type with its expected answer type by applying an algorithm proposed in the research by Breja and Jain (2018). (2) Question Reformulation which semantically parses why-question using NLTK tool to obtain noun phrases and content words useful for generating a query. Query is generated by applying Boolean AND and OR on content words and appending cue phrase 'because' that is input to Answer Candidate Extraction phase.

Answer Candidates Extraction: The reformulated query from Question reformulation phase is posed on web to obtain top-5 web pages. Further each web page is parsed to obtain answer passages containing maximum query terms, named entities and sentences containing causal cue phrases like 'because', 'due to', 'as a result', 'causes' etc. Thus these top-5 ranked passages are considered as a set of answer candidates which are further utilized for answer re-ranking, thus obtaining a possible combination of 5 possible answer candidates to each why-question.

Answer Re-Ranker: The answer candidates are re-ranked by analyzing the features comprising lexico-syntactic, semantic and contextual similarities between each question and answer candidate. Further answer re-ranker is trained and tested on weighted feature sets to achieve 0.64 MRR.

Answer Validation: This process matches the answer type of the answer candidate returned by the answer re-ranker module. The highest ranked answer with matched answer type is returned to the user.

DATA SETUP

A dataset of 1000 why-questions is gathered from different question answering sites like Yahoo! Answers (Yahoo! Answers, 2020), Quora (Quora, 2020) and Answers.com (Answers.com, 2020). With this, some questions are also collected from the dataset posted by Suzan Verberne on her personal webpage (Data download – Suzan Verberne, 2010). With each why-question, 5 answer candidates are retrieved through answer extraction process, thus resulting in 5000 why-QA pairs.

As our proposed work comprises training of dataset by different machine learning and deep learning techniques, there is a need of significant and appropriate dataset of why-QA. Since there are no existing appropriate sources of why-questions and their answers available, the exigency is to create a new dataset of why-questions with their corresponding answer candidates for the research. Concerning the appropriateness of the dataset, there is a significant number of why-questions and their answers. Since there are closely equal distribution of question and their expected answer type as determined from the algorithm given in Breja and Jain (2018), the dataset is free from any sampling bias.

Answer candidates to each why-question are ranked by 10 participants who are colleagues of different age group. Participants of different age group are selected to capture the variation in their understanding. Each participant has adopted closed card sorting algorithm to provide a rank on the scale of 1-5 (1 indicating the highest rank, 5 indicating the lowest rank) according to their knowledge and understanding (Spencer and Warfel, 2004).

Further a group technique is adopted where all participants collaborate together as a team and arrive at a conclusion with one appropriate rank to each answer candidate, thus achieving 0.82 Kappa value i.e. 82% inter-rater agreement (Widmann et al., 2020). That final rank is treated as a ground truth labeling of answer candidates utilized for calculating the value of MRR as discussed in section 7.

FEATURES USED FOR ANSWER RE-RANKING WITH THEIR RELEVANCE

This section discusses features utilized for answer re-ranking categorized on the basis of Natural Language Processing (NLP) phases. There are broadly four stages of NLP viz. Morphological Processing and Syntax analysis which determines the presence and order of lexical terms, Semantic analysis understands the meaning of a text, and pragmatic analysis which utilizes contextual features to determine the actual interpretation of a sentence. The paper extends the scope and covers all the dimensions of NLP to determine the importance of each phase in why-question answering.

Lexico-Syntactic Features:

These features uncover the lexical-syntactic similarity features determined between each question and answer candidate.

a) *Tf-idf Similarity:* TF-IDF quantifies the importance of each term in a document by utilizing Bag-of-word approach to determine the most relevant document for a query. The concept is utilized to find the relevant answer candidate to a question based on the importance of terms included in them.

Since the similarity is based on BOW approach, it couldn't capture the position of text in a document, semantics and terms co-occurrences.

It is calculated as:

$$tf-idf\left(t,a\right)=tf\left(t,a\right)*idf\left(t,a\right) \tag{1}$$

Where, tf(t,a)= frequency of term t in answer candidate a

idf(t,a)=log (no. of answer candidates possible for a question / no. of answer candidates with term t in it)

b) *Cosine Similarity:* Irrespective of the size of document, cosine similarity determines the document similarity by capturing the orientation of its angle. The key idea is to find the relatedness between each word occurring in question and corresponding answer candidate by representing them in vector space.

It is calculated by following formula:

$$cos\theta = \frac{\vec{q}.\vec{a}}{\|\vec{q}\|\|\vec{a}\|} = \frac{\sum_{1}^{n} q_i a_i}{\sqrt{\sum_{1}^{n} q_i^2}\sqrt{\sum_{1}^{n} a_i^2}} \tag{2}$$

Where, $\vec{q.a} = \sum_{1}^{n} q_i a_i = q_1 a_1 + q_2 a_2 + \ldots\ldots\ldots\ldots + q_n a_n$ denotes dot product of question and answer candidates vectors.

Cosine similarity overcomes the limitation of counting common words as used in Euclidean approach, thus determines the similarity by capturing many features between the words in documents. The angle between two vectors is directly proportional to the similarity. Since it is implemented using tf-idf vectorization, it is termed as 'cosine+tf-idf' similarity.

c) *Document similarity:* Document similarity is generally utilized to check plagiarism between documents where document text is represented as vectors and an angle between two document vectors are measured. Here vectors determine the frequency of words appearing in a document. The idea is to find the distance between question and answer candidates by determining the cosine angle between their vectors. Unlike cosine similarity, it doesn't find importance of words through tf-idf rather finds the count of words. Its value ranges from $0^{\circ} \text{to} 90^{\circ}$, where 0° denotes identical documents and 90° represents dissimilar documents.

d) *Jaccard Similarity:* Also termed as Jaccard index, determines the similarity between documents by measuring the count of common words out of total words. It is the ratio of intersection of documents over union of documents. Jaccard distance is the reciprocal of Jaccard index which is calculated as 1- Jaccard index. The range of Jaccard Similarity is 0 to 1. That is, if there are no common terms, Jaccard score is 0 and if they are identical, Jaccard score is 1.

$$Jaccard\left(question, anscandidate_i\right) = \sum_{i=1}^{5} \frac{question \cap anscandidate_i}{question \cup anscandidate_i} \tag{3}$$

e) *N-gram overlaps:* N-gram is a sequence of n terms from a document. N-gram overlap also termed as *'keywords in context'* determines the clustered keywords distance. It takes into account the phrases used in question and document, thus capturing the degree to which answer candidate having similar context as question.

It can be measured as:

$$sim_{ngrams}\left(p,q\right)=\frac{\sum_{\forall x\in P}h\left(x\right)\frac{1}{d\left(x,x_{max}\right)}}{\sum_{i=1}^{n}w_i} \tag{4}$$

Where, P: set of n-gram having highest weight in question q and included in answer candidate p, Q: set of n-grams in answer candidate p, n: total number of terms, h(x) is the summation of weights of terms in n-gram x (Buscaldi et al., 2010). The above set of lexico-syntactic features couldn't resolve the vocabulary mismatch problem. There is a possibility of lexical mismatch between question and answer candidate as answers to why-question mostly contain different terms but their meaning and context is appropriate. Thus, to reference such cases, semantic and contextual features are also considered.

Semantic Similarity Features:

a) *WordNet wup similarity:* WordNet is a lexical database composed of synsets of terms that incorporates meaning and different concepts of terms. WordNet::Similarity is a package utilized to implement the semantic similarity and relatedness between two concepts i.e question and answer candidates in this case. WuPalmer similarity (wup similarity) calculates the similarity between word senses by considering the depth of their synsets in hypernym tree along with the depth of LCS.

$$Wu-Palmer=2*\frac{depth\left(lcs\left(s_1,s_2\right)\right)}{depth\left(s_1\right)+depth\left(s_2\right)} \tag{5}$$

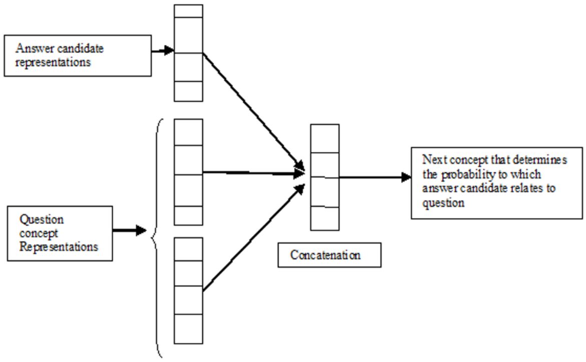
b) *WordNet Path Similarity:* This similarity captures the relatedness between noun and verb 'is-a' hierarchy i.e. it determines the similarity by considering the depth of shortest paths between the word synsets. Distance is inversely proportional to similarity or relatedness. The paper utilizes this concept to find the similarity between question and answer candidates by considering the word synsets present in them (Pedersen et al., 2004).

c) *Fuzzy matching:* This measure finds word-to-word correlations between phrases of question and answer candidates using the fuzzy set IR model which targets capturing syntax and semantics of concepts (Lee & Ng, 2007).

d) *Similarity based on semantic nets and corpus statistics:* Semantic-nets is one of the knowledge representation techniques in the form of graph which uncovers indirect relationships between terms used in question and their corresponding answer candidates. This feature captures inflectional knowledge where words appear in different forms but conveys similar meaning, thus resolving the lexical gap and considering the relevancy between question and answer candidates. During its implementation, Brown corpus of WordNet is utilized to incorporate the information content values of words (Li et al., 2006).

e) *Doc2vec:* In order to make the machines understand intention and context of a text, a sentence embedding technique is utilized to represent sentences in document with their semantics as vectors. Doc2vec is a concept introduced in 2014 as an extension to word2vec model. The model utilizes both the concepts of continuous bag-of-word model (CBOW) and continuous skip-gram model (CSG) to calculate Doc2vec similarity and determine the probability to which each answer candidate contributed positively to the question as illustrated in Figure 2 (Shperber, 2017).

Figure 2. Doc2vec model for question and its answer candidate


Answer candidate representations
Question concept Representations
Concatenation
Next concept that determines the probability to which answer candidate relates to question

f) *Word Mover Distance:* It is the most significant approach to retrieve the best relevant document corresponding to a query by determining the distance between two documents even if they have no terms in common. It employs skip-gram approach and targets both syntactic and semantic approach to find relevancy of answer candidate to a query (Kim et al., 2017).

g) *Universal Sentence Encoder:* It is one of the best sentence embedding techniques introduced by Google in 2018 based on deep learning framework. It comprises two encoder models viz. Transformer that utilizes self-attention technique and Deep Averaging Network that determines the average of unigram and bigram embeddings to feed into deep neural network and produces final sentence embedding of 512 dimensions (Cer et al., 2018). The concept of universal sentence encoder which inputs the sentence embeddings of question and their possible answer candidates to a deep neural network is illustrated in Figure 3.

Figure 3. Framework of Universal Sentence Encoder technique for finding relevancy

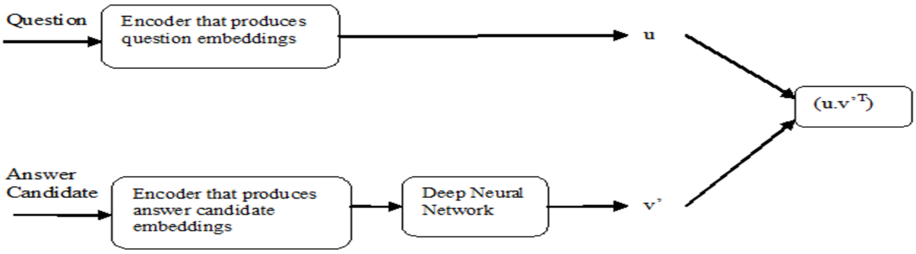
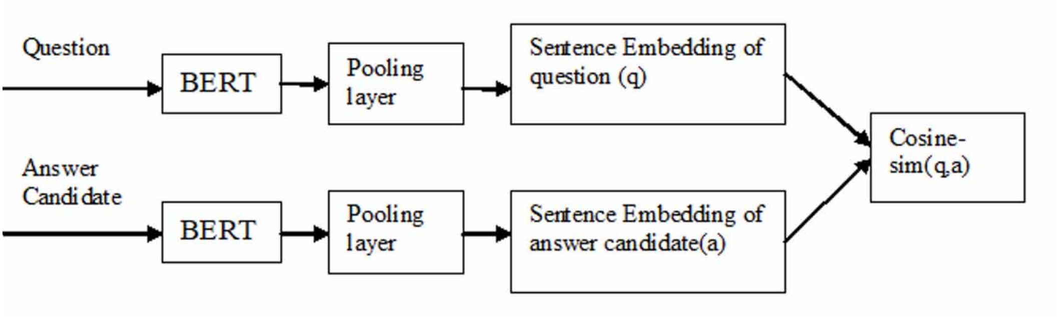

Question
Encoder that produces question embeddings
u
(u.v'T)
Answer Candidate
Encoder that produces answer candidate embeddings
Deep Neural Network
v'

Figure 4. Framework of BERT model for finding similarity between question and answer candidate


Question
Answer Candidate
BERT
BERT
Pooling layer
Pooling layer
Sentence Embedding of question (q)
Sentence Embedding of answer candidate(a)
Cosine-sim(q,a)

Contextual Features

a) *Soft-Cosine similarity:* Soft-cosine similarity captures the context of a sentence by determining the context of its words and thus utilizing it to find similarity between contextual vectors of question and their answer candidates. It employs FastText model for word embedding and is based on a corpus made from the words encountered in a question and their set of five answer candidates in the dataset (Sidorov et al., 2014).

b) *SBERT:* SentenceBERT is another sentence embedding technique introduced in 2018 which is a modification over pre-trained BERT networks utilized for evaluating the relevancy of question and its answer candidate. SBERT is based on the combination of siamese and triplet network architecture that takes query and its corresponding answer candidates as input which are passed into BERT model with a pooling layer to generate meaningful sentence embeddings. Further sentence embeddings are compared by measuring cosine similarity between each question and answer candidate that is illustrated in Figure 4. During the implementation, 'bert-base-nli-mean-tokens' is used as a pre-trained model (Kratzwald et al., 2019). SBERT is significantly applicable for semantic similarity comparisons and clustering tasks.

c) *InferSent cosine distance:* Infersent also termed as encoder-MLP is another sentence embedding technique introduced by Facebook in 2018 which is trained on Natural Language Interface dataset and categorized text as entailment, contradiction, and neutral. The model is utilized to capture the inferences between each question and answer candidate. Positive value depicts entailed relation, negative value depicts contradiction and zero value depicts no relation between question and answer candidate. It utilizes two versions of Infersent, one based on GlovE and another on FastText vectors (Huilgol, 2020).

d) *Sentiment Analysis:* This feature captures the word and phrase polarity of question and answer candidates to evaluate the semantic orientation between them, assuming that correct answer candidate to a positive sentiment question will be positive and vice versa.

Algorithm1: Finding MRR from each feature set

Input: Question Q_i and set of answer candidates (A_1,A_2,.....A_5).
Output: Determining role of feature set in answering questions
Steps: 1. For each QA pair (Q_i and A_i):

ALGORITHMS TO DETERMINE IMPORTANCE OF FEATURE SETS WITH THEIR RESULTS

This section proposed two algorithms (1) Finding MRR from each feature set and (2) Scoring each features according to their importance.

a) Proposed Algorithm to Find MRR from each Feature Set

The algorithm 1 finds Mean Reciprocal Rank from each feature type viz. lexico-syntactic, semantic and contextual features which helps to quantify the role in answering why-questions.

1.1. Find cosine, tf-idf, jaccard, n-gram overlaps, document similarity between Q_i and A_i
1.2. Find average value (avg1) of lexico-syntactic similarity between Q_i and A_i
1.3. Find wup, path, semantic_nets, doc2vec, universal_encoder, wmd similarity between Q_i and A_i
1.4. Find average (avg2) of semantic similarity between Q_i and A_i
1.5. Find value of soft-cosine, InferSent, BERT, sentiment analysis between Q_i and A_i
1.6. Find average (avg3) of contextual similarity between Q_i and A_i
1.7. Assign rank_lexical, rank_semantic, rank_contextual to each answer candidate based on the average values calculated above in Steps 1.2,1.4 and 1.6 respectively.
1.8. Compare the rank obtained from values with actually ranking provided by participants as discussed in Section 4.

As MRR targets rank of most relevant answer candidate, if the highest ranked answer candidate has position 1, it is assigned MRR 1, if the highest ranked answer candidate has position 3, it is assigned MRR 1/3 which in turn results MRR from lexico-syntactic features as 0.588, semantic features as 0.62 and contextual features to be 0.596. This clearly states that for finding correct answer to a why-question, semantic features place highest priority than contextual features which in turn higher than lexico-syntactic features.

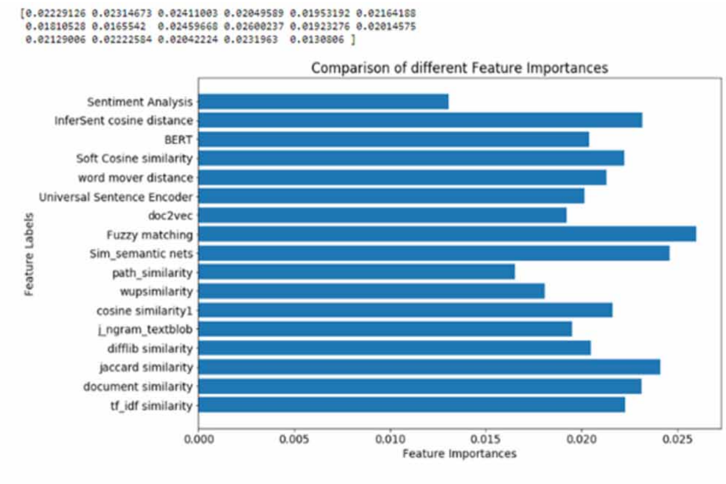
b) Finding Score of Features using Ensemble Technique

This section describes ExtraTreesClassifier ensemble model for finding out the features importance and assigning a score to each features that would help in learning how they play a vital role for re-ranking answer candidates, the results of which is depicted in Figure 5. ExtraTreesClassifier, better than RandomForest combines the result of multiple segregated decision trees to find the classification output. Features importance is determined by Gini index of the features. It gives best results in regression and classification tasks.

ALGORITHM FOR VALIDATING ANSWER CANDIDATES

This section proposes an algorithm 2 that validates answer candidates and returns one most accurate answer to a question by utilizing the concept of answer type matching.

Figure 5. Feature importance and their respective scores



```
[0.02229126 0.02314673 0.02411003 0.02049589 0.01953192 0.02164188
 0.01810528 0.0165542  0.02459668 0.02600237 0.01923276 0.02014575
 0.02129006 0.02222584 0.02042224 0.0231963  0.0130806 ]
```

Algorithm 2: Process of Answer Validation

Input: Question and its set of five answer candidates
Output: Most accurate answer candidate to a question
Given: Expected answer type of question and answer candidate using Algorithm proposed in Breja and Jain (2018)
Step1: For each (Q_i, A_i):
Find the average of all features i.e. lexico-syntactic, semantic and contextual similarities between Q_i and A_i
Step2: Arrange A_i in descending order of their similarity values
Step3: Match the answer type of highest ranked answer. If it matches expected answer type of a question, return that as an accurate answer
Step 4: Else return the next highest ranked answer having same answer type as of question

Table 1 briefs the performance of answer validation by implementing algorithm 2.

Table 1. Performance of answer validation

Rank of appropriate answer candidate with matched answer type	Performance
Rank 1	80%
Rank 2	15%
Rank 3 and Rank 4	5%

IMPLEMENTATION DETAILS

The work discussed in this paper has been implemented in Python 3.6. The values of lexico-syntactic, semantic and contextual features discussed in Section 6 are based on NLP concepts which are implemented using NLTK (Natural Language Toolkit) in Python. Tf-idf and cosine similarity are calculated by constructing document-term matrix and fitting the dataset to tf-idf and countvectorizer. WordNet path similarity is implemented by first tagging question and answer candidate by POS tag and converting Penn Treebank tag to WordNet tag. This process considers pos-tags noun, verb, adjective and adverb. Then similarity is calculated by determining the similarity value of each question term with the most similar term in the answer sentences. Word embeddings (Word2vec models) require installation of gensim 3.8.3. FastText model is pre-trained using 'fasttext-wiki-news-subwords-300' which is utilized for calculating soft-cosine similarity. The value of sentiment analysis is calculated using Vader (Valence Aware Dictionary and Sentiment Reasoner). Doc2vec is one of the gensim model trained using window size as 2 and 20 as vector_size which is used to determine the answer sentences contributed positively to the question. SBERT utilizes 'bert-base-nli-mean-tokens' as a pre-trained model to compute sentence embeddings of question and answer sentences. Universal Sentence encoder requires tensorflow hub and InferSent is implemented by PyTorch which utilizes 'glove.840B.300d' as pre-trained GloVe word vectors (Srinivasa-Desikan, 2018).

COMPARISION WITH OTHER WORKS

Table 2 below compares the performance of research works on answer re-ranker of Non-factoid question answering systems.

CONCLUSION

The paper presents an approach to re-rank answer candidates of a why-question through feature selection methods. Various features covering lexical-syntactic, semantic and contextual similarities have been employed to find the relevancy of each answer candidate to a question. Following this, ExtraTreesClassifier is utilized to find the scoring of each feature. The results of feature importance depict the importance of semantic features over contextual and lexico-syntactic features for appropriate answering why-questions thus achieving a performance metric of 0.64 MRR. Further answer candidates are validated with the answer type determined from Question classification phase. Highest ranked answer candidate with matched answer type is returned to the user, thus achieving an accuracy of 80% answer candidates at rank 1, 15% at rank 2, and remaining 5% from rank 3 and rank 4.

There is still a scope to incorporate discourse processing depending on answer type and employing common sense reasoning concepts to further improve the results. Further the discussed algorithms can be applied for restricted domain question answering system to analyze the impact on their performance.

Table 2. Comparison of our proposed work with other research works

Reference	Domain and type of Questions	Technique utilized	Significance of techniques	Lexico-syntactic features	Semantic features	Contextual features	Discourse Features	Performance Metric
(Verberne et al., 2010)	Open-domain English why-type	Machine learning technique for learning to rank and their cost functions trained on 37-feature sets	Features representing similarity between question and answer sets capturing number of words overlap between question focus and answer titles. Different learning to rank approaches are applied to visualize the best results	✔	✗	✗	✗	0.35 MRR
(Jansen & Surdeanu, 2014)	Open-domain English why and how-type	Discourse based models integrated with lexical-semantic models	Intra and inter-sentential discourse features help to capture the context and relations between sentences of answer candidate with lexical semantics capturing the words with meaning of answer to a question	✔	✔	✗	✔	0.49 MRR
(Molino et al. 2014)	Open-domain English How-type	Vector representation of words for semantic matching between questions and answers	Vector representation to capture the context of different words and utilized linguistic and text quality features to capture overlapping of words with the quality of information in answer candidate with respect to the question	✗	✔	✗	✗	0.7909 MRR
(Fried & Jansen 2015)	Open-domain English How-type	Higher order lexical-semantic models	Higher order methods utilized to learn indirect associations between question and answer words with lexical-semantic models to overcome lexical chasm problem and compute semantic similarity between question and answer candidates	✔	✔	✗	✗	0.5396 MRR
(Tran & Niederee 2018)	Restricted domain (Insurance and Financial domain) English non-factoid	Attention-based deep learning neural network models	To learn low-dimensional linguistic similarities and concentrate on semantic relevance between question and answer vectors	✗	✔	✔	✗	0.762 and 0.616 MRR using Siamese Architecture mlp
(Higashinaka & Isozaki 2008)	Open-domain Japanese why-type	Corpus based approach to automatically collect causal expressions and utilize content similarity and causal relations as features to train re-ranker	Corpus based approach for automatically collecting causal expressions to overcome the limitation of handcrafted causal patterns. Causality is the crucial part in answering why-questions by capturing cause-effect relations	✔	✔	✗	✗	0.305 MRR

continued on next page

Table 2. Continued

Reference	Domain and type of Questions	Technique utilized	Significance of techniques	Lexico-syntactic features	Semantic features	Contextual features	Discourse Features	Performance Metric
(Oh et al., 2012)	Open-domain Japanese why-type	Supervised classifiers trained and tested on features representing comprising morpho-syntactic, semantic word classes and sentiment analysis	Morpho-syntactic analysis to capture associations by identifying n-grams of morphemes, word phrases and syntactic dependencies. Semantic word classes to capture semantic word classes associations between questions and answers with sentiment analysis to identify semantic orientation and polarity between question and answers	✔	✔	✗	✗	0.604 MAP
(Sakamoto, 2015)	Open-domain Japanese non-factoid	Graph based model with an extended HITS algorithm	Graph based model helped to capture different viewpoints of answer with question and HITS algorithm to capture the relatedness between different answer fragments	✔	✗	✗	✗	0.575 MRR
(Oh et al., 2013)	Open-domain Japanese why-type	Capturing intra and inter-sentential causal relations	None of the researches have addressed inter-sentential causal relations, thus this method tries to identify the answer boundary trained by causal relations	✔	✔	✗	✗	0.41 MAP
Our Proposed Approach	Open-domain English why-type	Machine learning techniques trained on features capturing lexico-syntactic, semantic and contextual features	Integrates syntactic, semantic and contextual for finding similarity between question and its corresponding answer candidates and judging the impact of each feature set for why-type question answering	✔	✔	✔	✗	0.64 MRR

REFERENCES

Answers.com. (n.d.). Retrieved September 28, 2020, from https://www.answers.com/

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

Buscaldi, D., Rosso, P., Gómez-Soriano, J. M., & Sanchis, E. (2010). Answering questions with an n-gram based passage retrieval engine. *Journal of Intelligent Information Systems*, *34*(2), 113–134. doi:10.1007/s10844-009-0082-y

Cer, D., Yang, Y., Kong, S. Y., Hua, N., Limtiaco, N., St. John, R., Constant, N., Guajardo-Céspedes, M., Yuan, S., Tar, C., Sung, Y. H., Strope, B., & Kurzweil, R. (2018). Universal sentence encoder for English. *EMNLP 2018 - Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Proceedings,* 169–174.

Fried, D., Jansen, P., Hahn-Powell, G., Surdeanu, M., & Clark, P. (2015). Higher-order lexical semantic models for non-factoid answer reranking. *Transactions of the Association for Computational Linguistics*, *3*, 197–210. doi:10.1162/tacl_a_00133

Hashimoto, C., Torisawat, K., De Saeger, S., Oh, J. H., & Kazama, J. (2012). Excitatory or inhibitory: A new semantic orientation extracts contradiction and causality from the web. *EMNLP-CoNLL 2012 - 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 619–630.

Higashinaka, R., & Isozaki, H. (2008). Corpus-based question answering for why-questions. *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I.*

Huilgol, P. (2020). *Top 4 Sentence Embedding Techniques using Python!* https://www.analyticsvidhya.com/blog/2020/08/top-4-sentence-embedding-techniques-using-python/

IBM Watson (n.d). Retrieved September 28, 2020, from https://www.ibm.com/in-en/watson

Jansen, P., Surdeanu, M., & Clark, P. (2014). Discourse complements lexical semantics for non-factoid answer reranking. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (*Volume 1*: Long Papers)* (pp. 977–986). doi:10.3115/v1/P14-1092

Kim, S., Fiorini, N., Wilbur, W. J., & Lu, Z. (2017). Bridging the gap: Incorporating a semantic similarity measure for effectively mapping PubMed queries to documents. *Journal of Biomedical Informatics*, *75*, 122–127. doi:10.1016/j.jbi.2017.09.014 PMID:28986328

Kratzwald, B., Eigenmann, A., & Feuerriegel, S. (2019). Rankqa: Neural question answering with answer re-ranking. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 6076–6085.

Lee, J. W., & Ng, Y. K. (2007). Using fuzzy-word correlation factors to compute document similarity based on phrase matching. *Proceedings - Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007, 2*, 186–191.

Li, Y., McLean, D., Bandar, Z. A., O'Shea, J. D., & Crockett, K. (2006). Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, *18*(8), 1138–1150. doi:10.1109/TKDE.2006.130

Liu, T. Y. (2011). Learning to rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, *3*(3), 225–231. doi:10.1561/1500000016

Molino, P., & Aiello, L. M. (2014). Distributed representations for semantic matching in non-factoid question answering. *CEUR Workshop Proceedings*, *1204*, 38–45.

Oh, J. H., Torisawa, K., Hashimoto, C., Kawada, T., De Saeger, S., Kazama, J., & Wang, Y. (2012). Why question answering using sentiment analysis and word classes. *EMNLP-CoNLL 2012 - 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Proceedings of the Conference*, 368–378.

Oh, J. H., Torisawat, K., Hashimoto, C., Sano, M., De Saeger, S., & Ohtake, K. (2013). Why-question answering using intra- And inter-sentential causal relations. *ACL 2013 - 51st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 1, 1733–1743.

Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004). WordNet: Similarity-Measuring the Relatedness of Concepts. In AAAI (Vol. 4, pp. 25-29). Academic Press.

Quora. (n.d.). Retrieved September 28, 2020, from https://www.quora.com/

Sakamoto, K., Nagao, K., Kobayashi, H., Shibuki, H., Mori, T., & Kando, N. (2015). *Re-ranking answer candidates based on exhaustiveness of variety of answer viewpoints in non-factoid QA*. Academic Press.

Shperber, G. (2017, July 26). *A Gentle Introduction to Doc2Vec*. Wisio. https://medium.com/wisio/a-gentle-introduction-to-doc2vec-db3e8c0cce5e

Sidorov, G., Gelbukh, A., Gómez-Adorno, H., & Pinto, D. (2014). Soft similarity and soft cosine measure: Similarity of features in vector space model. *Computación y Sistemas*, *18*(3), 491–504. doi:10.13053/cys-18-3-2043

Spencer, D., & Warfel, T. (2004). Card sorting: a definitive guide. *Boxes and arrows*. https://boxesandarrows.com/card-sorting-a-definitive-guide

Srinivasa-Desikan, B. (2018). *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras*. Packt Publishing Ltd.

Surdeanu, M., Ciaramita, M., & Zaragoza, H. (2011). Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics*, *37*(2), 351–383. doi:10.1162/COLI_a_00051

Tran, N. K., & Niederée, C. (2018). A Neural Network-based Framework for Non-factoid Question Answering. *Companion Proceedings of the Web Conference 2018,* 1979–1983.

Verberne, S. (2010). *Data Download*. https://liacs.leidenuniv.nl/~verbernes/wordpress/research/data-download/

Verberne, S., Van Halteren, H., Theijssen, D., Raaijmakers, S., & Boves, L. (2010). Learning to Rank QA Data: Evaluating Machine Learning Techniques for Ranking Answers to Why-Questions. *Proceedings of the Workshop Learning to Rank for Information Retrieval 2009*, 41-48.

Widmann, M. (2020). *Cohen's Kappa: What It Is, When to Use It, and How to Avoid Its Pitfalls.* https://thenewstack.io/cohens-kappa-what-it-is-when-to-use-it-and-how-to-avoid-its-pitfalls/

Yahoo. Answers. (n.d.). Retrieved September 28, 2020, from https://in.answers.yahoo.com/

Manvi Breja is a PhD Scholar from National Institute of Technology, Kurukshetra. She has completed her M.Tech from YMCA University, Faridabad, India. She has 6 years of experience in teaching and research. Her area of interest is Information Retrieval, Data Mining and Natural Language Processing.

Sanjay Kumar Jain, PhD (2006) & M.Tech (MNNIT, Allahabad, India. He is a Professor in the Department of Computer Engineering at NIT Kurukshetra, India. He is actively involved in research and has 27 years of experience of teaching and research. His current research areas include database, data mining, data science and requirements/software engineering.