AN EFFICIENT PASSAGE RANKING TECHNIQUE FOR A QA SYSTEM

Pooja A[1], Vinodh Krishnan[2], Geetha Manjunath[3]

(1) Amazon Development Centre India, Bangalore
(2) Oracle India Private Ltd, Bangalore
(3) HP Labs, India, Bangalore
apooja@amazon.com, krishnan.vinodh@gmail.com,
geetha.manjunath@hp.com

ABSTRACT

Question answering (QA) systems provide an intuitive way of requesting concise information from a given data source. An important stage of such a system is the passage ranking stage, which ranks the possible answers based on their relevance to the question. There has been a lot of previous work on passage ranking, employing lexical, semantic or syntactic methods, but to our knowledge there has been no method that comprehensively combines all 3 features. In this paper, we present a passage ranking technique that leverages lexical, semantic and syntactic features together to rank the answers efficiently and effectively. This paper highlights the differences and improvements of the proposed technique over existing state-of-the-art techniques like SSTK and IBM Model. The passage ranking technique has been evaluated with TREC QA dataset and is observed to give a significant 26.5% improvement in MRR over the existing state-of-the-art SSTK technique.

KEYWORDS

Question Answering, Factoid, Information Retrieval, Passage Ranking, Unstructured Data

1. INTRODUCTION

Unlike search engines, an automatic Question Answering (QA) system has the unique advantage of delivering precise reliable information for a user query without any requirement on the part of the user to do any post-processing. A recent user study [1] conducted at HP Labs, India showed that even non-tech savvy users would find it more comfortable to use the web if they can obtain precise answers to their queries on their mobile devices.

Over the past few years, many techniques have been proposed to develop question answering systems targeted to answer queries posed on different domains using various data sources. However, there are areas where there is still some scope for improvement. Below are some of the challenges in building an efficient automated QA System:

- Understanding the context and intent of the question posed in natural language.
- Determining the appropriate answer source for answering the question posed.
- Designing robust techniques to rank relevant data extracted from structured and unstructured content to determine the answer for the given question.

DOI : 10.5121/ijcsit.2012.4606

In this paper, we propose a technique to tackle the third challenge. The proposed technique does this by considering lexical, semantic and syntactic features to determine the passage from the corpus that most likely contains the answer to the posed question. The lexical features are obtained by considering the n-grams of the terms present in question/relevant data. The semantic features are generated using NLP tools and knowledge bases like WordNet and Wikipedia. The syntactic features are derived from the parse tree of question/relevant data. Finally, the answer extraction from the ranked data is performed by using answer type matches as well as syntactic matches. The technique is shown to work efficiently with a factoid based QA System but can be extended to subjective QAs as well, as the features considered are not specific to factoid Questions and Passages. The techniques it has been compared to are also not Factoid specific and can be employed in Complex QA Systems as well (such as the SSTK shown in this paper[3]).
The rest of the paper is organized as follows: Section 2 provides the related work. Section 3 gives the overview of the Question Answering System used, Section 4 details passage ranking technique and Section 5 gives evidence to the claim made through experimental results.

2. RELATED WORK

There has been extensive work in the field of question answering systems. However, we concentrate on prior work related to passage ranking. A survey [2] of all the state-of-the-art lexical passage ranking algorithms like Mitre, IBM, bm25, etc. indicates the scope for improvement in them and also notes that the passage retrieval performance depends on document retriever as well. Paper [3] describes and compares the various methods of passage ranking right from the initial n-gram and tf-idf measures to more recent Syntactic and Semantic measures like the SSTK[4] in the context of complex question answering.

With respect to this paper, the SSTK[4] and AType-DP-IP[5] techniques are the most relevant ones as both of these methods consider all the three category of features that we consider. It is to be noted that most of the work prior to these considered a subset of the features and hence, will be inferior in performance as we show in the experiments section. Works like [6], use semantic and syntactic techniques, but do not consider lexical techniques such as mismatch and inverse passage frequency (ipf). Other techniques like [7], don't consider semantic similarities and employ the use of ontologies like Wordnet[8]. SSTK[4] is a kernel function proposed for automatic question categorization and it incorporates syntactic dependencies and term similarity based on WordNet[8]. The same kernel is used for Question Answering [3] and is shown to outperform prior passage ranking techniques. A more recent technique, AType-DP-IP[5], ranks the passages after aligning the syntactic structures based on the question's answer type and detected named entities in the candidate passage. Synonymy between question and passage terms is also considered while ranking. Further, the parse tree path matches are weighted by scores learnt using a training set of question-answer pairs. This involves an additional step of developing a relevant corpus of question-answer pairs.

In this paper, the features considered by above papers are retained and few more important ones are added to enhance the performance further. The proposed technique considers the degree of mismatch as well for passage scoring similar to IBM metric[9], and paper[10]. However these methods consider semantic and mismatch but do not consider syntactic techniques, which is shown to perform better [3] while the proposed method combines all the three. In addition, we use ipf (inverse passage frequency), which has been introduced in [11] as inverse sentence frequency, as term weights for scoring parse tree path matches. Using ipf based weighting captures the underlying characteristics of the data source and, at the same time, is in coherence with our desire to develop an unsupervised ranking to enable handling of domain independent questions. Further, the match in the context of usage is weighed more in the final scoring function ensuring that the highest ranking passages are highly relevant to the question. We also filter passages based on n-

gram matches before extensive ranking is performed. No such lenient filtering is performed by prior work, which causes all the irrelevant sentences to be considered in the ranking phase, increasing the overall computation time. Thus, the proposed technique can be considered as a logical extension of SSTK[4] or AType-DP-IP[5] for unsupervised ranking.

3. QUESTION ANSWERING SYSTEM

At the macro level the QA System we employed, as shown in Figure 1, consists of Question Analyser, Data Source Identifier, Data Processor and Answer Extractor.

Question Analyser

The Question Analyser comprises of NLP tools like Parts-Of-Speech (POS) tagger, Stemmer and Stop words remover to extract the focus of the question i.e. the part of the question crucial for getting the answer. In the case of factoid question answering, it involves extraction of data required to reach current data source (Target) and the specific information queried for (Property). In addition, the expected answer type (EAT) of the question is determined through rules based on common constructs used for posing questions like Person (Who), Location (Where) and Time (When). If the answer type cannot be determined through the nature of the Question noun (like in some complex questions), it is assigned as Other. This keeps the accuracy of answer type recognition high enabling the passage ranker to use answer types effectively in scoring passages.

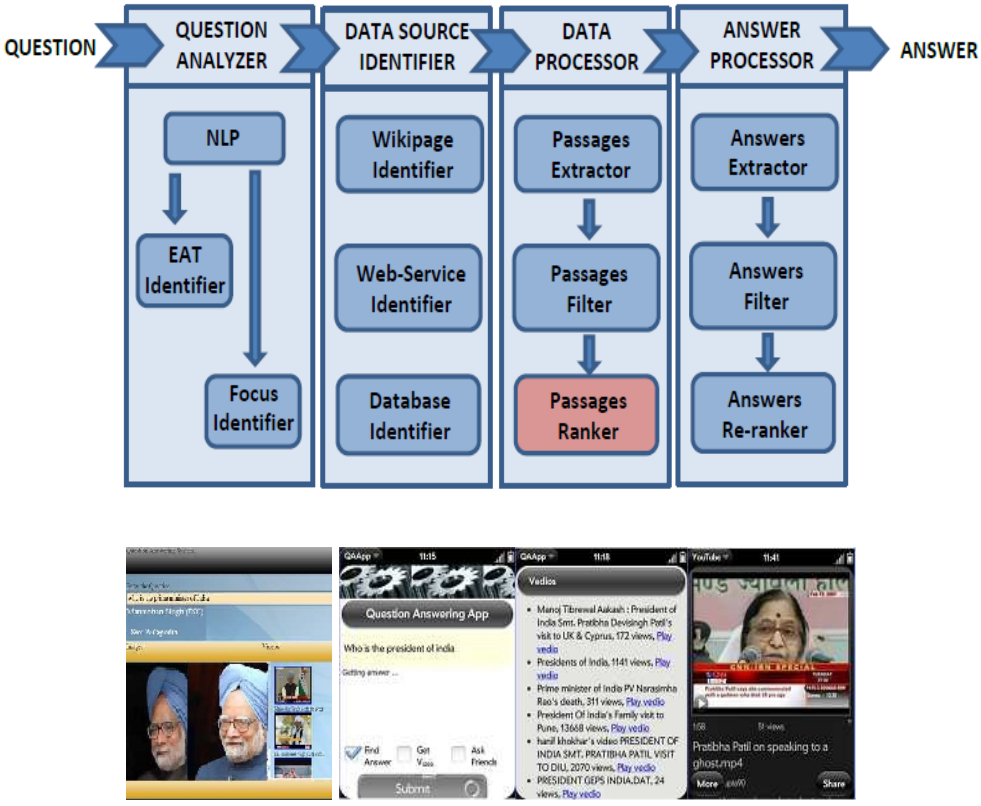

Figure 1 – Block Diagram of the QA System along with the screen shots of output on desktop and on mobile device simulator

Data Source Identifier

In the Data Source Identifier stage, the appropriate data source is identified using the Target and if needed, the Property of the question. Though the system can be easily expanded to handle database and other web sources, the present implementation uses the entire Wikipedia as its data corpus and through Target, the relevant wiki-page is located. The selected wiki-page is forwarded to the Data Processor stage.

Data Processor

At the Data Processor stage, the passages are obtained from paragraphs of wiki-page by the Passage Extractor. Passages can be obtained from a given paragraph by splitting it at sentence level or at a coarser level with the other extreme being to consider the entire paragraph as a single passage. Since this QA system concentrates on factoid questions, the answers of which are few words long, we consider each sentence as a passage. Thus obtained passages are scored by the proposed Passage Ranker using various features. The passage with highest score gets the top rank whereas the lowest scored passage gets the lowest rank.

Answer Extractor

Finally, the scored passages are passed to the Answer Extractor. The answer can be extracted from the top ranked sentences by matching the expected answer type of the question with the named entities in the sentences. When the expected answer type is 'Other' the answer is identified by matching the category of the subject word of the question (determined using Wikipedia) with the named entities. The identified answer can be passed to any search engine to augment the answer with corresponding images and videos. The named entities in the sentence can be identified using any standard Named Entity Recognizer.

4. PASSAGE RANKING

4.1 Basic Concepts Used

In this section, we explain the knowledge bases and terminologies relevant to the proposed passage ranking technique.

WordNet

WordNet [8] is a large lexical database of English. It comprises of nouns, verbs, adjectives and adverbs of English grouped into cognitive synonyms (synsets) and connected based on different relations to form a network of meaningfully related words and concepts that can be navigated. The prominent relations used to form the network involves super-subordinate relation i.e. hypernymy, hyponymy or ISA relation, part-whole relation i.e. meronymy, synonym relation and antonym relation. In this paper, synonym relations are used to capture the semantics of the question and the data extracted.

Parse Tree

A parse tree of a string, in general, is an ordered, rooted tree representing its syntactic structure according to predefined formal grammar. In this paper, the linguistic parse tree of a given sentence is considered and is constructed based on English grammar. We used Stanford Parser for constructing linguistic parse tree, referred simply as 'parse tree' in the rest of the paper. In Figure

2, we can see the parse tree of "Who is the prime minister of India?" as generated by Stanford Parser is provided.

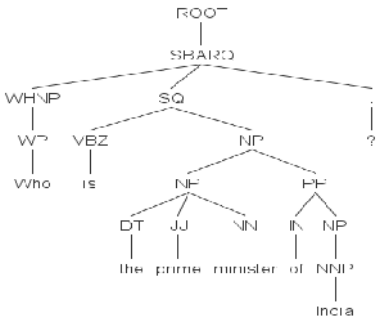

Figure 2: Parse tree of "Who is the prime minister of India"

As seen from Figure 2, a parse tree comprises of the words of the given sentence as the leaf nodes with the branch nodes indicating the grammatical relations between the words and the root node indicating the sentence itself.

Further, it can be observed that the branch nodes which are immediate ancestors of the leaf nodes (called pre-terminal nodes) are nothing but the POS tags of the term in the leaf nodes. In this paper, we refer to these special set of branch nodes as 'context nodes' as they indicate the context in which the term in the leaf node is used i.e. is the term used as a noun or verb or adjective etc. The parse tree is used to capture the syntax of question or data.

4.2 Features Chosen

Like SSTK[4] and AType-DP-IP[5], lexical and semantic features are used in conjunction with the structural similarity between question and given passages. Parse trees are developed for the question and for each of the answer sentences. As the tree is traversed, semantic and lexical features are identified and used to score the passage with respect to the question.

Under lexical category, we consider term match and mismatch. Like [11] and unlike SSTK[4] and AType-DP-IP[5], we use ipf (inverse passage frequency) as term weights. As desired, the ipf weighting ensures that the passages that contain the most important and distinguishing terms (non-frequent terms across passages) of the questions are given higher score. Using ipf weights instead of weights learnt using training corpus is in coherence with our desire to develop an unsupervised ranking to enable handling of domain independent factoid questions.

Like IBM metric[9] and unlike SSTK[4] and AType-DP-IP[5], the proposed technique separates out the mismatch component from the implicit mismatch factor usually introduced by normalization of passage score by length of the passage. A negative score is awarded to every question word (excluding stop words) that is present in the question but not in the answer. Among the variants tried, we found the use of L2 normalization with separate mismatch scoring component to give the best results.

Under semantic features, WordNet[8] synonym matches are considered. We use synonyms for matching question terms with passage terms and in addition, for matching expected answer type with type of named entities in passage like in AType-DP-IP[5]. Thus, for 'Who is the prime minister of India?'; the leaf node corresponding to 'Who' will contain synonyms of 'person' as well to help more flexible matching with named entity type.

Further, the term matches and synonym matches with match in the context of usage is weighed more in the final scoring function, as such matches are strong evidences of the passage being an answer to given question. This is a measure that perfectly combines the syntactic and semantic aspect of a sentence and contributes greatly to the score as seen in the results. This is an enhancement not previously given its deserved importance.

Under syntactic features, the structural similarity between question and given passage is used as the feature. Parse trees are developed for the question and for each of the answer sentences. As the tree is traversed, semantic and lexical features are identified and are used to score the relevance of the passage to the question. Context nodes, as already mentioned are also obtained from the parse tree of the candidate answer sentence.

Thus the overall formula is a weighted combination of these scores, perfectly integrating all the three measures. The weights are obtained through careful analysis.

4.3 Ranking Methodology

This subsection lists the steps involved in passage ranking.

4.3.1 Filtering

To begin with, the passages are filtered based on the n-gram matches. All the passages with at least a single term match with the question are retained. This lenient filtering ensures that the passages to be considered are reduced before the computationally intensive scoring is performed without losing much in ranking accuracy. No such lenient filtering is performed by prior work, which causes all the irrelevant sentences to be considered in the ranking phase, increasing the overall computation time.

4.3.2 Parse Trees with Semantic Spreading

After filtering, the first step in the scoring process is to obtain the parse tree for the given question and for the passages as well. Then, in the question parse tree, each leaf node containing non-stop, non-question indicating term is linked with the synonyms of that term. Further, the leaf node corresponding to the question indicating term in the parse tree is linked with the synonyms of the EAT. Thus, for ‘Who is the prime minister of India?’ the leaf node corresponding to ‘Who’ will contain synonyms of ‘person’ as well.

4.3.3 Stemming

Further, the leaf node terms of the question parse tree and passage parse tree are stemmed. Stemming is performed to accommodate for any tense or word form related variations.

4.3.4 Scoring

Scoring is done based on similarities between the parse tree of the candidate answer and the question under various different criteria as given below. In the component scoring functions that follow, s(.) indicates the set of synonyms of a given term excluding the term itself, t(.) provides the production rule/type of the given parse tree node, p(.) provides the parent node of the given node and w(.) provides the word/term corresponding to the given node. w(.) is valid only over the leaf nodes and t(.) is nothing but POS tags when the function operates on pre-terminal nodes (immediate ancestors of leaf nodes). While n denotes a node in the parse tree, L indicates the set of all leaf nodes in the parse tree and $\overline{L}$ indicates the complement of that set, i.e. all the nodes

other than the leaf nodes in the parse tree. Subscripts q and p are used to indicate parameters of the question and passage respectively.

IDF/IPF Weighting: In every category of scoring (except type match scoring), the score of the term is weighed by its inverse passage frequency. When the term occurs less frequently in the answer bank, the ipf value will be higher as it is indirectly proportional to the frequency of word occurrence. Similarly, when the term is more common in the corpus, the ipf weight is lower. This ensures that the rare terms in the question with respect to the corpus of possible answers are given higher weights as against the terms that appear in a lot of passages of the candidate answer bank. Thus, the passages that contain most important and distinguishing terms (non-frequent terms) of the questions are given higher scores. This is analogous to the inverse sentence frequency introduced in [11].

Exact Term Matching + Context Matching: Two components of the scoring function, $I_{e+c}(n_p,n_q)$ and $I_{e+nc}(n_p,n_q)$, are used to explicitly score the exact term matches between the question and the answer. These functions operate on the leaf nodes of the corresponding parse trees and their immediate ancestors. Under the exact match situation, two cases arise – the case where the context of usage also matches and the case where it does not match. The inclusion of context matching is something that is not given much importance in previous works. The context of usage is captured by the immediate ancestors i.e. the POS Tag nodes of the leaf nodes in the parse tree. When the term matches and the context also matches, the indicator function $I_{e+c}(n_p,n_q)$ operates. Else, $I_{e+nc}(n_p,n_q)$ is activated. We see later that $I_{e+c}(n_p,n_q)$ is weighted higher than $I_{e+nc}(n_p,n_q)$ in the final ranking function as desired.

$$I_{e+c}(n_p,n_q)=\begin{cases} idf(w(n_p)), if\ n_p \in L_p \,\&\, n_q \in L_q \,\&\, t(p(n_p)) = t(p(n_q)) \,\&\, w(n_p) = w(n_q) \\ 0, otherwise \end{cases}$$

$$I_{e+nc}(n_p,n_q)=\begin{cases} idf(w(n_p)), if\ n_p \in L_p \,\&\, n_q \in L_q \,\&\, w(n_p) = w(n_q) \,\&\, t(p(n_p)) \neq t(p(n_q)) \\ 0, otherwise \end{cases}$$

Synonym Matching and Context Matching: Here, we consider the case where, for a given term in the question, there is no exact matching term in the passage. Instead, there is a match at the synonym level. As in the case of exact match, there exists two cases – the one where the context of usage also matches and the other where the context does not match. While $I_{s+c}(n_p,n_q)$ denotes the first case, $I_{s+nc}(n_p,n_q)$ denotes the latter case. The negative values (-1 and -2) are introduced as mismatch scores in the expression since the word itself is not appearing in the candidate answer passage, but its synonym is present.

$$I_{s+c}(n_p,n_q)=\begin{cases} idf(w(n_p)) - 1, if\ n_p \in L_p \,\&\, n_q \in L_q \,\&\, w(n_p) \in s(n_q) \,\&\, t(p(n_p)) = t(p(n_q)) \\ 0, otherwise \end{cases}$$

$$I_{s+nc}(n_p,n_q)=\begin{cases} idf(w(n_p)) - 2, if\ n_p \in L_p \,\&\, n_q \in L_q \,\&\, w(n_p) \in s(n_q) \,\&\, t(p(n_p)) \neq t(p(n_q)) \\ 0, otherwise \end{cases}$$

Mismatch Score: $I_{mm}(n_p,n_q)$ is used to reduce the score for mismatches. A mismatch happens when a term in the question and a term in the passage appear completely unrelated i.e. the terms that do not match and are not present in each other's synonym set.

$$I_{mm}(n_p, n_q) = \begin{cases} 1, \text{ if } n_p \in L_p \,\&\, n_q \in L_q \,\&\, w(n_p) \neq w(n_q) \,\&\, w(n_p) \notin s(n_q) \,\&\, w(n_q) \notin s(n_p) \\ 0, \text{otherwise} \end{cases}$$

Type Matching: $I_t(n_p,n_q)$ is used to capture matching at the root and branch node levels. This captures the syntactic similarity between the question and the answer.

$$I_t(n_p, n_q) = \begin{cases} 1, \text{ if } n_p \in \overline{L}_p \,\&\, n_q \in \overline{L}_q \,\&\, t(n_p) = t(n_q) \\ 0, \text{otherwise} \end{cases}$$

By linearly combining all these individual score contributors, the score of the passage is obtained as given below,

$$S_p = \sum_{n_p \in T_p} \sum_{n_q \in T_q} S(n_p, n_q)$$

$$S(n_p, n_q) = \alpha I_{e+c}(n_p, n_q) + \beta I_{e+nc}(n_p, n_q) + \eta I_{s+c}(n_p, n_q) + \varepsilon I_{s+nc}(n_p, n_q) + \phi I_t(n_p, n_q) - \lambda I_{mm}(n_p, n_q)$$

The coefficients were chosen by validating different combinations on a sample of the TREC question set (that was excluded from the test set) to get the best possible performance. The values were, α =5, β =2, η =1, ε =2, ϕ =1, λ =2.

After the scoring is done for each passage/sentence, it is normalized by dividing it with the square root of the length of that sentence. The sentences are then ranked in decreasing order of their scores. The answer can then be extracted from the top ranked sentences using the EAT match with the named entities in the sentences.

5. EXPERIMENTS

For evaluating the proposed passage ranking technique performance with that of the existing state-of-the-art techniques, we considered TREC 2006 factoid questions dataset (TREC, 2006)[12], the dataset typically used for benchmarking. TREC 2006 dataset [12] comprises of factoid, list and other questions related to 75 targets. We used Wikipedia as the data source to answer them instead of the standard news dataset as web is the target source for the proposed QA system. We considered the whole of Wikipedia and used the target of each question that is provided in the dataset to locate the relevant wiki page. This formed the list for the candidate answer passages to the question (the corpus for the passages). A small fraction of the factoid question set has been removed as the information to answer these was not available in Wikipedia. After this, 322 of them were picked for evaluation. WordNet[8] was used to get synonyms of the words. Stanford CoreNLP (Klein and Manning, 2003) was used for obtaining the parse trees. We evaluated the proposed passage ranking technique against prominent prior existing techniques like n-gram matching, IBM metric [9] and SSTK[4]. Each of these techniques was run with the same corpus as that of the proposed technique, i.e. the passages from the relevant wiki page. To bring out the significance of different factors of the technique, we also compare it against its own variants, namely

1. n-gram matching with mismatch penalty (n-gram_Mismatch)
2. parse tree matching without ipf-weighting (PT_Simple)
3. PT_Simple with synonym matching and ipf-weighting (PT_Semantic)
4. PT_Semantic with mismatch penalty (PT_Mismatch)
5. PT_Mismatch with context matching (PT_Proposed)

For evaluation, we used the answer patterns created by Ken Litkowski available on TREC's site as golden standard data. However, for the time dependent questions, the answers were updated as per the 2011 version of wiki page. A passage was considered correct if it contained the golden standard answer. MRR (Mean Reciprocal Ranking) was used as the performance measure to compare these techniques. From Figure 3, it is clear that PT_Proposed has a significant MRR improvement compared to the baseline n-gram method (around 2 times higher). The MRR score, itself, indicates that the proposed technique can ensure correct answer to be in Top 2 (1/0.62 = 1.61) more often than SSTK does (1/0.49=2.04). Also, it is can be observed that each additional scoring component adds to the improvement in the passage ranking.

Let use consider the question "Who is the present host of the Daily Show?" taken from the TREC 2006 Dataset [12]. The answer passage, "Jon Stewart, the current host took over in January 1999" was the top ranked passage in our proposed method, whereas in SSTK[4] and PT_Semantic the top ranked passage was "The Daily show premiered on July 21, 1996, and was hosted by Craig Kilborn until December 1998" which is incorrect as it does not give the current host. Because these methods did not consider mismatch, the top ranked answer passage just returned a host, and not the present host. On the other hand, lexical methods like IBM that considers the mismatch and synonyms apart from direct matching failed to get the highest MRR as lot of times the answer returned would be just the one having more number of occurrences of question words in the answer. Further, the usage of context match also boosts the MRR score significantly (0.51 to 0.62) in the proposed technique as proven by the below example.

Q: Which is the headquarters of Nissan? (A question picked from present TREC 2006 Dataset)
The two close contenders for this question from Wikipedia were:

A1 : As of 2011, Nissan's global headquarters is located in Nishi-ku, Yokohama.
A2 : Nissan North America relocated its headquarters from Gardena, California to the Nashville, Tennessee area in July 2006.

Of these two, the top ranked answer picked by our method was the first one. It got the highest score because headquarters and Nissan were used in the same context (object and subject) as in the question. Also, the number of mismatches for this was smaller compared to the second candidate answer.

Consider the sample question "What was Irving Berlin's first big hit? " Given below are the top 5 answers from the proposed method and the other methods we have compared it to, with the passage source as Wikipedia (as of December 2011).

Table 1: Top ranked passages of Proposed Technique, SSTK[4] and IBM[9] to the given question

Rank	Proposed Technique	IBM Method[9]	SSTK[4]
1	His first hit song "alexander's ragtime band" became world famous.	It was irving berlin who was the very first to have created a real inherent american music	It was irving berlin who was the very first to have created a real inherent american music
2	One of the key songs that berlin wrote in his transition from ragtime to lyrical ballads was "a pretty girl is like a melody" which was considered one of berlin's "first big guns" according to historian alec wilder.	Irving berlin was the first to free the american song from the nauseating sentimentality which had previously characterized it and by introducing and perfecting ragtime he had actually given us the first germ of an american musical idiom	His first hit song "alexander's ragtime band" became world famous
3	It was irving berlin who was the very first to have created a real inherent american music	It was first sung by harry richman in 1930 and became a #1 hit and in 1939 clark gable sang it in the movie "idiot's delight.	One of the key songs that berlin wrote in his transition from ragtime to lyrical ballads was "a pretty girl is like a melody" which was considered one of berlin's "first big guns" according to historian alec wilder.
4	It was first sung by harry richman in 1930 and became a #1 hit and in 1939 clark gable sang it in the movie "idiot's delight.	One of the key songs that berlin wrote in his transition from ragtime to lyrical ballads was "a pretty girl is like a melody" which was considered one of berlin's "first big guns" according to historian alec wilder.	It was first sung by harry richman in 1930 and became a #1 hit and in 1939 clark gable sang it in the movie "idiot's delight.
5	Irving berlin was the first to free the american song from the nauseating sentimentality which had previously characterized it and by introducing and perfecting ragtime he had actually given us the first germ of an american musical idiom	His first hit song "alexander's ragtime band" became world famous.	This waltz-time hit went to #2 with rudy vallee and in 1937 reached #1 with tommy dorsey.

As seen, the top ranked answer in the proposed method "His first hit song 'alexander's ragtime band' became world famous." sufficiently and accurately answers the question whereas the top ranked passages of IBM and SSTK are not the correct and perfect answers to this question. This can be attributed to the improvements of the proposed technique over these methods.

As we can see through the results and examples, the proposed method performs better than the state-of-the-art passage ranking techniques like SSTK and IBM. Additionally MRR@10 and MRR@5 were calculated for the proposed method, and the values obtained were, 0.608 and 0.598 indicating that most answers were in the top 10 and top 5.

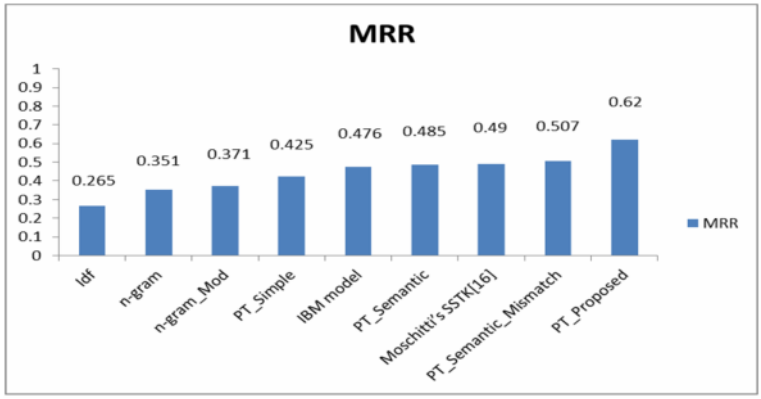

Figure 3 – MRR Scores for various passage ranking techniques against the proposed.

6. CONCLUSION

In this paper, we concentrated on passage ranking techniques for question answering systems and proposed novel variants. We considered Lexical, Semantic and Syntactic features together to come up with a comprehensive Passage Ranking Technique for a factoid based QA system which can be extended to non-factoid ones as well. Through experiments and examples, we have justified the addition of each feature in our Passage Ranking Technique. We proved the efficiency of the proposed technique over other techniques by evaluating against the TREC 2006 dataset. The proposed Passage Ranking technique is shown to have a MRR of 0.62 (Higher the MRR, better is the ranking) and is seen to outperform existing popular techniques, thus, making it a promising technique to pursue. Going further, we would like to productize the solution and measure its performance in real world scenario. In this regard, faster variants of this technique need to be explored.

ACKNOWLEDGEMENTS

We thank A. Varun Prakash for developing the initial version of QA System code with NLP units in place and Maria Kumar Pamisetty for helping with efficient implementation of QA system.

REFERENCES

[1] Mathur, N. and Sharma, V. (2009). Towards designing intent framework for analyzing information needs. HP TechReport.

[2] Tellex et al. (2003). Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering. In ACM SIGIR Conference on Research and Development in Information Retrieval, pages 41-47.

[3] Chali, Y., Joty, S. R., and Hasan, S. A. (2009). Complex Question Answering: Unsupervised Learning Approaches and Experiments. In International Journal of Computational Intelligence Research. 35: pages 1-47.
[4] Bloehdorn, S. and Moschitti, R. (2007). Combined Syntactic and Semantic Kernels for Text Classification. In European Conference on Information Retrieval. 4425: pages 307-318.
[5] Aktolga, E., Allan, J. and Smith, D. A. (2011). Passage Reranking for Question Answering using Syntactic Structures and Answer Types. In European Conference on Information Retrieval. 6611: pages 617-628.
[6] R. Sun, J. Jiang, Y. F. Tan, H. Cui, T.-S. Chua, and M.-Y. Kan. 2006. Using syntactic and semantic relation analysis in question answering. In Proc. TREC 2005.
[7] M. W. Bilotti, E. Nyberg. Improving text retrieval precision and answer accuracy in question answering systems, In Proc. 2nd IRQA Coling Workshop, Stroudsburg, PA, 2008, pp. 1–8
[8] Miller, G. 1995. WordNet: A Lexical Database for English. In Communications of the ACM. 38(11): pages 39-41.
[9] Ittycheriah, A., Franz, M., and Roukos, S. (2001). IBM's statistical question answering system—TREC-10. In Proceedings of the Tenth Text REtrieval Conference.
[10] V. Murdock and W. B. Croft. Simple translation models for sentence retrieval in factoid question answering. In Proc. SIGIR Workshop on Information Retrieval for Question Answering, pages 31--35, 2004.
[11] Momtazi, S., Lease, M., Klakow, D. Effective Term Weighting for Sentence Retrieval. In ECDL (2010) 482-485
[12] TREC Dataset (2006). http://trec.nist.gov/data/qa/2006_qadata/qa.06.guidelines.html#main

Authors

Pooja A is currently working as a Machine Learning Scientist with Amazon, India. She previously worked at HP Labs, India as a research consultant where she, along with the other authors developed the proposed method for passage ranking in QA Systems. Her research interests include Machine Learning and Natural Language Processing.

Vinodh Krishnan is currently working at Oracle India as a Software Developer. He previously interned at HP Labs during the course of his undergraduate study at BITS Pilani. His research interests include Natural Language Processing, Artificial Intelligence and Machine Learning.

Geetha Manjunath is a Senior Research Scientist and Master Technologist at HP Labs, India and mentored the development of the proposed method. She has developed many innovative solutions and published papers in the area of Embedded Systems, Java Virtual Machine, Storage Virtualization and Semantic Web. She is currently leading a research project on Simplifying Web for non-tech savvy users. Before joining HP, she was a senior technical member at Centre for Development of Advanced Computing (C-DAC), Bangalore for 7 years - where she lead a research team to develop parallel compilers for distributed memory machines. She is a gold medallist from Indian Institute of Science where she did her Masters in Computer Science in 1991 and is currently pursuing her PhD. She was awarded the TR Shammanna Best Student award from Bangalore University in her Bachelor's degree for topping across all branches of Engineering. She holds four US patents with one more pending grant.