Answering Questions from Multiple Documents – the Role of Multi-Document Summarization

Pinaki Bhaskar
Department of Computer Science & Engineering,
Jadavpur University, Kolkata – 700032, India
pinaki.bhaskar@gmail.com

Abstract

Ongoing research work on Question Answering using multi-document summarization has been described. It has two main sub modules, document retrieval and Multi-document Summarization. We first preprocess the documents and then index them using Nutch with NE field. Stop words are removed and NEs are tagged from each question and all remaining question words are stemmed and then retrieve the most relevant 10 documents. Now, document graph-based query focused multi-document summarizer is used where question words are used as query. A document graph is constructed, where the nodes are sentences of the documents and edge scores reflect the correlation measure between the nodes. The system clusters similar texts from the graph using this edge score. Each cluster gets a weight and has a cluster center. Next, question dependent weights are added to the corresponding cluster score. Top two-ranked sentences of each cluster is identified in order and compressed and then fused to a single sentence. The compressed and fused sentences are included into the output summary with a limit of 500 words, which is presented as answer. The system is tested on data set of INEX QA track from 2011 to 2013 and best readability score was achieved.

1 Introduction

With the explosion of information in Internet, Natural language Question Answering (QA) is recognized as a capability with great potential. Traditionally, QA has attracted many AI researchers, but most QA systems developed are toy systems or games confined to laboratories and to a very restricted domain. Several recent conferences and workshops have focused on aspects of the QA research. Starting in 1999, the Text Retrieval Conference (TREC)[1] has sponsored a question-answering track, which evaluates systems that answer factual questions by consulting the documents of the TREC corpus. A number of systems in this evaluation have successfully combined information retrieval and natural language processing techniques. More recently, Conference and Labs of Evaluation Forums (CLEF)[2] are organizing QA lab from 2010.

INEX[3] has also started Question Answering track. INEX 2011 designed a QA track (SanJuan et al., 2011) to stimulate the research for real world application. The Question Answering (QA) task is contextualizing tweets, i.e., answering questions of the form "what is this tweet about?" INEX 2012 Tweet Contextualization (TC) track gives QA research a new direction by fusing IR and summarization with QA. The first task is to identify the most relevant document, for this a focused IR is needed. And the second task is to extract most relevant passages from the most relevant retrieved documents. So an automatic summarizer is needed. The general purpose of the task involves tweet analysis, passage and/or XML elements retrieval and construction of the answer, more specifically, the summarization of the tweet topic.

Automatic text summarization (Jezek and Steinberger, 2008) has become an important and timely tool for assisting and interpreting text information in today's fast-growing information age. An Abstractive Summarization ((Hahn and Romacker, 2001) and (Erkan and Radev, 2004)) attempts to develop an understanding of the main concepts in a document and then expresses those concepts in clear natural language. Extractive Summaries (Kyoomarsi et al., 2008) are formu-

[1] http://trec.nist.gov/
[2] http://www.clef-initiative.eu//
[3] https://inex.mmci.uni-saarland.de/

Proceedings of the Student Research Workshop associated with RANLP 2013, pages 14–21,
Hissar, Bulgaria, 9-11 September 2013.

lated by extracting key text segments (sentences or passages) from the text, based on statistical analysis of individual or mixed surface level features such as word/phrase frequency, location or cue words to locate the sentences to be extracted. Our approach is based on Extractive Summarization.

In this paper, we describe a hybrid Question Answering system of document retrieval and multi-document summarization. The document retrieval is based on Nutch[4] architecture and the multi-document summarization system is based graph, cluster, sentence compression & fusion and sentence ordering. The same sentence scoring and ranking approach of Bhaskar and Bandyopadhyay (2010a and 2010b) has been followed. The proposed system was run on the data set of three years of INEX QA track from 2011 to 2013.

2 Related Work

Recent trend shows hybrid approach of question answering (QA) using Information Retrieval (IR) can improve the performance of the QA system. Schiffman et al. (2007) successfully used methods of IR into QA system. Rodrigo et al. (2010) removed incorrect answers of QA system using an IR engine. Pakray et al. (2010) used the IR system into QA and Pakray et al. (2011) proposed an efficient hybrid QA system using IR.

Tombros and Sanderson (1998) presents an investigation into the utility of document summarization in the context of IR, more specifically in the application of so-called query-biased summaries: summaries customized to reflect the information need expressed in a query. Employed in the retrieved document list displayed after retrieval took place, the summaries' utility was evaluated in a task-based environment by measuring users' speed and accuracy in identifying relevant documents.

A lot of research work has been done in the domain of both query dependent and independent summarization. MEAD (Radev et al., 2004) is a centroid based multi document summarizer, which generates summaries using cluster centroids produced by topic detection and tracking system. NeATS (Lin and Hovy, 2002) selects important content using sentence position, term frequency, topic signature and term clustering. XDoX (Hardy et al., 2002) identifies the most salient themes within the document set by passage clustering and then composes an extraction summary, which reflects these main themes. Graph-based methods have been also proposed for generating summaries. A document graph-based query focused multi-document summarization system has been described by Paladhi et al. (2008) and Bhaskar and Bandyopadhyay (2010a and 2010b).

In the present work, we have used the IR system as described by Pakray et al. (2010 and 2011) and Bhaskar et al. (2011) and the automatic summarization system as discussed by Bhaskar and Bandyopadhyay (2010a and 2010b) and Bhaskar et al. (2011).

3 System Architecture

In this section the overview of the system framework of the current INEX system has been shown. The current INEX system has two major sub-systems; one is the Focused IR system and the other one is the Automatic Summarization system. The Focused IR system has been developed on the basic architecture of Nutch, which use the architecture of Lucene[5]. Nutch is an open source search engine, which supports only the monolingual Information Retrieval in English, etc. The Higher-level system architecture of the combined Tweet Contextualization system of Focused IR and Automatic Summarization is shown in the Figure 1.

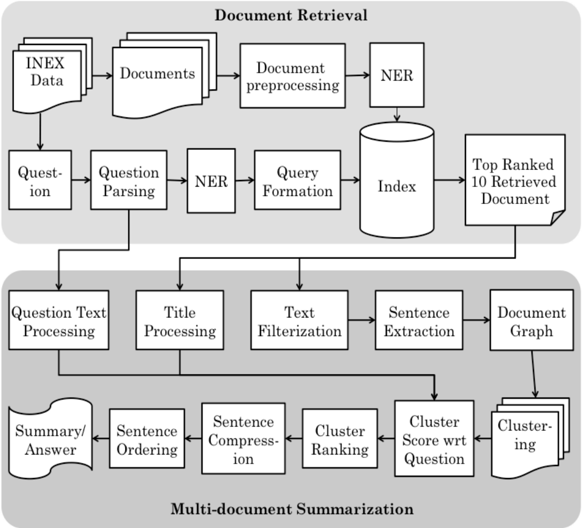

Figure 1. Higher-level system architecture

4 Document Retrieval

4.1 Document Parsing and Indexing

The web documents are full of noises mixed with the original content. In that case it is very diffi-

[4] http://nutch.apache.org/

[5] http://lucene.apache.org/

cult to identify and separate the noises from the actual content. INEX 2012 corpus had some noise in the documents and the documents are in XML tagged format. So, first of all, the documents had to be preprocessed. The document structure is checked and reformatted according to the system requirements.

XML Parser: The corpus was in XML format. All the XML test data has been parsed before indexing using our XML Parser. The XML Parser extracts the Title of the document along with the paragraphs.

Noise Removal: The corpus has some noise as well as some special symbols that are not necessary for our system. The list of noise symbols and the special symbols like "&quot;", "&amp;", "''", multiple spaces etc. is initially developed manually by looking at a number of documents and then the list is used to automatically remove such symbols from the documents.

Named Entity Recognizer (NER): After cleaning the corpus, the named entity recognizer identifies all the named entities (NE) in the documents and tags them according to their types, which are indexed during the document indexing.

Document Indexing: After parsing the documents, they are indexed using Lucene, an open source indexer.

4.2 Question Parsing

After indexing has been done, the questions had to be processed to retrieve relevant documents. Each question / topic was processed to identify the question words for submission to Lucene. The questions processing steps are described below:

Stop Word Removal: In this step the question words are identified from the questions. The stop words[6] and question words (what, when, where, which etc.) are removed from each question and the words remaining in the questions after the removal of such words are identified as the question tokens.

Named Entity Recognizer (NER): After removing the stop words, the named entity recognizer identifies all the named entities (NE) in the question and tags them according to their types, which are used during the scoring of the sentences of the retrieved document.

Stemming: Question tokens may appear in inflected forms in the questions. For English, standard Porter Stemming algorithm[7] has been used to stem the question tokens. After stemming all the question tokens, queries are formed with the stemmed question tokens.

4.3 Document Retrieval

After searching each query into the Lucene index, a set of retrieved documents in ranked order for each question is received.

First of all, all queries were fired with AND operator. If at least ten documents are retrieved using the query with AND operator then the query is removed from the query list and need not be searched again. If not then the query is fired again with OR operator. OR searching retrieves at least ten documents for each query. We always ranked the retrieved document using AND operator higher than the same using OR operator. Now, the top ranked ten relevant documents for each question is considered for milti-document summarization. Document retrieval is the most crucial part of this system. We take only the top ranked ten relevant documents assuming that these are the most relevant documents for the question from which the query had been generated.

5 Multi-Document Summarization

5.1 Graph-Based Clustered Model

The proposed graph-based multi-document summarization method consists of following steps:

(1) The document set $D = \{d_1, d_2, \ldots d_{10}\}$ is processed to extract text fragments, which are sentences in this system as it has been discussed earlier. Let for a document d_i, the sentences are $\{s_{i1}, s_{i2}, \ldots s_{im}\}$. Each text fragment becomes a node of the graph.

(2) Next, edges are created between nodes across the documents where edge score represents the degree of correlation between inter-documents nodes.

(3) Seed nodes are extracted which identify the relevant sentences within D and a search graph is built to reflect the semantic relationship between the nodes.

(4) Now, each node is assigned a question dependent score and the search graph is expanded.

(5) A question dependent multi-document summary is generated from the search graph.

Each sentence is represented as a node in the graph. The text in each document is split into

[6] http://members.unine.ch/jacques.savoy/clef/

[7] http://tartarus.org/~martin/PorterStemmer/java.txt

sentences and each sentence is represented with a vector of constituent words. If pair of related document is considered, then the inter document graph can be represented as a set of nodes in the form of bipartite graph. The edges connect two nodes corresponding to sentences from different documents.

Construct the Edge and Calculate Edge Score: The similarity between two nodes is expressed as the edge weight of the bipartite graph. Two nodes are related if they share common words (except stop words) and the degree of relationship can be measured by equation 1 adapting some traditional IR formula (Varadarajan and Hristidis, 2006).

$$\text{Edge\_Score} = \frac{\sum_{w\in(t(u)\cap t(v))} ((tf(t(u),w)+tf(t(v),w))\times idf(w))}{size(t(u))+size(t(v))} \quad (1)$$

where, *tf*(*d* , *w*) is number of occurrence of *w* in *d*, *idf* (*w*) is the inverse of the number of documents containing *w*, and *size*(*d*) is the size of the documents in words. Actually for a particular node, total edge score is defined as the sum of scores of all out going edges from that node. The nodes with higher total edge scores than some predefined threshold are included as seed nodes.

But the challenge for multi-document summarization is that the information stored in different documents inevitably overlap with each other. So, before inclusion of a particular node (sentence), it has to be checked whether it is being repeated or not. Two sentences are said to be similar if they share for example, 70% words in common.

Construction of Search Graph: After identification of seed/topic nodes a search graph is constructed. For nodes, pertaining to different documents, edge scores are already calculated, but for intra document nodes, edge scores are calculated in the similar fashion as said earlier. Since, highly dense graph leads to higher search / execution time, only the edges having edge scores well above the threshold value might be considered.

5.2 Identification of Sub-topics through Markov Clustering

In this section, we will discuss the process to identify shared subtopics from related multi source documents. We already discussed that the subtopics shared by different news articles on same event form natural (separate) clusters of sentences when they are represented using document graph. We use Markov principle of graph clustering to identify those clusters from the document graph as described by Bhaskar and Bandyopadhyay (2010b).

The construction of question independent part of the Markov clusters completes the document-based processing phase of the system.

5.3 Key Term Extraction

Key Term Extraction module has two sub modules, i.e., question term extraction and Title words extraction.

Question Term Extraction: First the question is parsed using the Question Parsing module. In this Question Parsing module, the Named Entities (NE) are identified and tagged in the given question using the Stanford NER[8] engine. The remaining words after stop words removal are stemmed using Porter Stemmer.

Title Word Extraction: The titles of each retrieved documents are extracted and forwarded as input given to the Title Word Extraction module. After removing all the stop words from the titles, the remaining tile words are extracted and used as the keywords in this system.

5.4 Question Dependent Process

The nodes of the already constructed search graph are given a question dependent score. Using the combined scores of question independent score and question dependent score, clusters are reordered and relevant sentences are collected from each cluster in order. Then each collected sentence has processed and compressed removing the unimportant phrases. After that the compressed sentences are used to construct the summary.

Recalculate the Cluster Score: There are three basic components in the sentence weight like question terms, title words and synonyms of question terms dependent scores, which are calculated using equation 2.

$$w = \sum_{t=1}^{n_t}(n_t - t + 1)\left(\sum_p\left(1 - \frac{f_p^t - 1}{N_i}\right)\right)\times b \quad (2)$$

where, w is the term dependent score of the sentence i, t is the no. of the term, n_t is the total no. of term, f_p^t is the possession of the word which was matched with the term t in the sentence i, N_i is the total no. of words in sentence i and b is boost factor of the term, which is 3, 2 or 1 for question terms, title words and synonyms respectively. These three components are added to get the final weight of a sentence.

Recalculate the Cluster Ranking: We start by defining a function that attributes values to

[8] http://www-nlp.stanford.edu/ner/

the sentences as well as to the clusters. We refer to sentences indexed by i and question terms indexed by j. We want to maximize the number of question term covered by selection of sentences:

$$maximize \sum_j w_j^q q_j \quad (3)$$

where, w_j^q is the weight of question term j in the sentence i and q_j is a binary variable indicating the presence of that question term in the cluster. We also take the selection over title words and synonyms of the question terms. We collect the list of synonyms of the each word in the questions from the WordNet 3.0[9]. The general sets of tile words and synonyms are indexed by k and l respectively. So we also want to maximize the number of title words and synonyms covered by a selection of sentences using similar calculation like question terms using equation 3.

So, the question dependent score of a cluster is the weighted sum of the question terms it contains. If clusters are indexed by x, the question dependent score of the cluster x is:

$$c_x^q = \sum_{i=1}^{n} \sum_j w_j^q q_j + \sum_{i=1}^{n} \sum_k w_k^t t_k + \sum_{i=1}^{n} \sum_l w_l^s s_l \quad (4)$$

where, c_x^q is the question dependent score of the cluster x, n is the total no. of sentences in cluster x. Now, the new recalculated combined score of cluster x is:

$$c_x = c_x^g + c_x^q \quad (5)$$

where, c_x is the new score of the cluster x and c_x^g is the question independent cluster score in the graph of cluster x. Now, all the clusters are ranked with their new score c_x.

5.5 Retrieve Sentences for Summary

Get the highest weighted two sentences of each cluster, by the following equation:

$$\max\left(\sum_j w_j^q q_j + \sum_k w_k^t t_k + \sum_l w_l^s s_l\right) \forall i \quad (6)$$

where, i is the sentence index of a cluster. The original sentences in the documents are generally very lengthy to place in the summary. So, we are actually interested in a selection over phrases of sentence. After getting the top two sentences of a cluster, they are split into multiple phrases. The Stanford Parser[10] is used to parse the sentences and get the phrases of the sentence.

5.6 Sentence Compression

All the phrases which are in one of those 34 relations in the training file, whose probability to drop was 100% and also do not contain any question term, are removed from the selected summary sentence as described by Bhaskar and Bandyopadhyay (2010a). Now the remaining phrases are identified from the parser output of the sentence and search phrases that contain at least one question term then those phrases are selected. The selected phrases are combined together with the necessary phrases of the sentence to construct a new compressed sentence for the summary. The necessary phrases are identified from the parse tree of the sentence. The phrases with `nsubj` and the `VP` phrase related with the `nsubj` are some example of necessary phrases.

5.7 Sentence Selection for Summary

The compressed sentences for summary have been taken until the length restriction of the summary is reached, i.e. until the following condition holds:

$$\sum_i l_i S_i < L \quad (7)$$

where, l_i is the length (in no. of words) of compressed sentence i, S_i is a binary variable representing the selection of sentence i for the summary and L (=100 words) is the maximum summary length. After taking the top two sentences from all the clusters, if the length restriction L is not reached, then the second iteration is started similar to the first iteration and the next top most weighted sentence of each cluster are taken in order of the clusters and compressed. If after the completion of the second iteration same thing happens, then the next iteration will start in the same way and so on until the length restriction has been reached.

5.8 Sentence Ordering and Coherency

In this paper, we will propose a scheme of ordering; in that, it only takes into consideration the semantic closeness of information pieces (sentences) in deciding the ordering among them. First, the starting sentence is identified which is the sentence with lowest positional ranking among selected ones over the document set. Next for any source node (sentence) we find the summary node that is not already selected and have (correlation value) with the source node. This node will be selected as next source node in ordering. This ordering process will continue until the nodes are totally ordered. The above ordering scheme will order the nodes independent of the

[9] http://wordnet.princeton.edu/

[10] http://nlp.stanford.edu/software/lex-parser.shtml

actual ordering of nodes in the original document, thus eliminating the source bias due to individual writing style of human authors. Moreover, the scheme is logical because we select a sentence for position *p* at output summary, based on how coherent it is with the (p-1)th sentence. The main sentence's number has been taken as the sentence number of the fused sentence.

Now the generated multi-document summary is presented as the answer of the corresponding question.

6 Experiment Result

The proposed system has been tested on the data set of INEX QA track from 2011 to 2013.

6.1 Informative Content Evaluation

The Informative Content evaluation (SanJuan et al., 2011) by selecting relevant passages using simple log difference of equation 8 was used:

$$\sum \log\left(\frac{\max\left(P\left(t/reference\right),P\left(t/summary\right)\right)}{\min\left(P\left(t/reference\right),P\left(t/summary\right)\right)}\right) \quad (8)$$

The year wise evaluation scores of informativeness of all topics are shown in the figure 2.

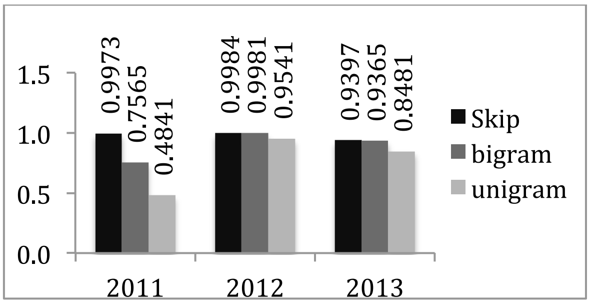

Figure 2. The evaluation scores of Informativeness by organizers of all topics

6.2 Readability Evaluation

For Readability evaluation (SanJuan et al., 2011) all passages in a summary have been evaluated according to Syntax (S), Soundness/Anaphora (A), Redundancy (R) and Relevancy/Trash (T). If a passage contains a syntactic problem (bad segmentation for example) then it has been marked as Syntax (S) error. If a passage contains an unsolved anaphora then it has been marked as Anaphora (A) error. If a passage contains any redundant information, i.e., an information that have already been given in a previous passage then it has been marked as Redundancy (R) error. If a passage does not make any sense in its context (i.e., after reading the previous passages) then these passages must be considered as trashed, and readability of following passages must be assessed as if these passages were not present, so they were marked as Trash (T). The readability evaluation scores are shown in the figure 3. Our relaxed metric i.e relevancy (T) score is the best score and strict metric i.e average of non redundancy (R), soundness (A) and syntax (S) score is the 4th best score among all the runs from all the participants of INEX 2011.

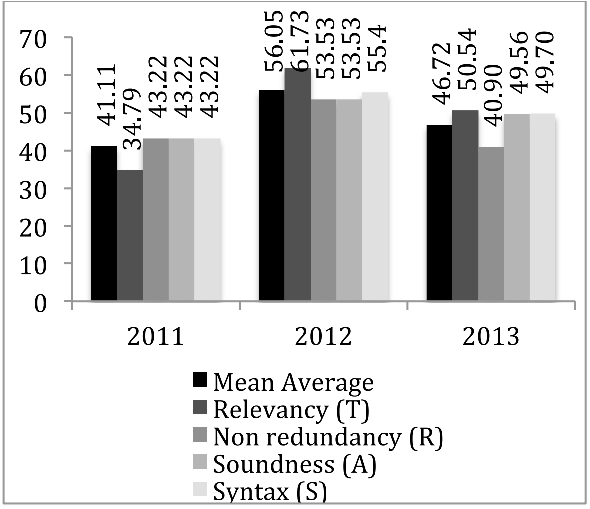

Figure 3. The evaluation scores of Readability Evaluation

7 Discussion

The tweet question answering system has been developed and tested on the data set of the Question Answering (QA) / Tweet Contextualization (TC) track of the INEX evaluation campaign from 2011 to 2013. The overall system has been evaluated using the evaluation metrics provided as part of the QA/TC track of INEX. Considering that the system is completely automatic and rule based and run on web documents, the evaluation results are satisfactory as readability scores are very high and in the relaxed metric we got the highest score of 43.22% in 2011, which will really encourage us to continue work on it in future.

Future works will be motivated towards improving the performance of the system by concentrating on co-reference and anaphora resolution, multi-word identification, para-phrasing, feature selection etc. In future, we will also try to use semantic similarity, which will increase our relevance score.

Acknowledgments

We acknowledge the support of the DeitY, MCIT, Govt. of India funded project "Development of Cross Lingual Information Access (CLIA) System Phase II".

References

Álvaro Rodrigo, Joaquın Pérez-Iglesias, Anselmo Peñas, Guillermo Garrido, and Lourdes Araujo. 2010. *A Question Answering System based on Information Retrieval and Validation*, In: CLEF 2010 Workshop on Multiple Language Question Answering (MLQA 2010), Padua, Italy.

Anastasios Tombros, and Mark Sanderson. 1998. *Advantages of Query Biased Summaries in Information Retrieval*. In: the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 2-10, ISBN:1-58113-015-5, ACM New York, USA.

Barry Schiffman, Kathleen McKeown, Ralph Grishman, and James Allan. 2007. *Question Answering using Integrated Information Retrieval and Information Extraction*. In: HLT/NAACL 07, pp. 532-539, Rochester, NY, USA.

Bidhan Chandra Pal, Pinaki Bhaskar, and Sivaji Bandyopadhyay. 2011. *A Rule Base Approach for Analysis of Comparative and Evaluative Question in Tourism Domain*. In: 6th Workshop on Knowledge and Reasoning for Answering Questions (KRAQ'11) in IJCNLP 2011, pp 29-37, Thailand.

Chin-Yew Lin, and Eduard Hovy. 2002. *From Single to Multidocument Summarization: A Prototype System and its Evaluation*. In: ACL, pp. 457-464.

Dragomir R. Radev, Hongyan Jing, Małgorzata Stys, Daniel Tam. 2004. *Centroid- based summarization of multiple documents*. In: Information Processing and Management. 40, pp. 919–938.

Eric SanJuan, V´eronique Moriceau, Xavier Tannier, Patrice Bellot, and Josiane Mothe. 2011. *Overview of the INEX 2011 Question Answering Track (QA@INEX)*. In: Focused Retrieval of Content and Structure, 10th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX), Geva, S., Kamps, J., Schenkel, R. (Eds.). Lecture Notes in Computer Sc., Springer.

Farshad Kyoomarsi, Hamid Khosravi, Esfandiar Eslami, and Pooya Khosravyan Dehkordy. 2008. *Optimizing Text Summarization Based on Fuzzy Logic*. In: Seventh IEEE/ACIS International Conference on Computer and Information Science, pp. 347--352. IEEE, University of Shahid Bahonar Kerman, UK.

Gu¨ne ̧s Erkan, and Dragomir R. Radev. 2004. *LexRank: Graph-based Centrality as Salience in Text Summarization*. In: Journal of Artificial Intelligence Research, vol. 22, pp. 457-479.

Hilda Hardy, Nobuyuki Shimizu, Tomek Strzalkowski, Liu Ting, G. Bowden Wise, and Xinyang Zhang. 2002. *Cross-document summarization by concept classification*. In: the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 121-128, ISBN: 1-58113-561-0, ACM New York, NY, USA.

Karel Jezek, and Josef Steinberger. 2008. *Automatic Text summarization*. In: Snasel, V. (ed.) Znalosti 2008. ISBN 978-80-227-2827-0, pp.1--12. FIIT STU Brarislava, Ustav Informatiky a softveroveho inzinierstva.

Partha Pakray, Pinaki Bhaskar, Santanu Pal, Dipankar Das, Sivaji Bandyopadhyay, and Alexander Gelbukh. 2010. *JU_CSE_TE: System Description QA@CLEF 2010 – ResPubliQA*. In: CLEF 2010 Workshop on Multiple Language Question Answering (MLQA 2010), Padua, Italy.

Partha Pakray, Pinaki Bhaskar, Somnath Banerjee, Bidhan Chandra Pal, Alexander Gelbukh, and Sivaji Bandyopadhyay. 2011. *A Hybrid Question Answering System based on Information Retrieval and Answer Validation*. In: Question Answering for Machine Reading Evaluation (QA4MRE) at CLEF 2011, Amsterdam.

Pinaki Bhaskar, and Sivaji Bandyopadhyay. 2010a. *A Query Focused Multi Document Automatic Summarization*. In: the 24th Pacific Asia Conference on Language, Information and Computation (PACLIC 24), pp 545-554, Tohoku University, Sendai, Japan.

Pinaki Bhaskar, and Sivaji Bandyopadhyay. 2010b. *A Query Focused Automatic Multi Document Summarizer*. In: the International Conference on Natural Language Processing (ICON), pp. 241--250. IIT, Kharagpur, India.

Pinaki Bhaskar, Amitava Das, Partha Pakray, and Sivaji Bandyopadhyay. 2010c. *Theme Based English and Bengali Ad-hoc Monolingual Information Retrieval in FIRE 2010*. In: the Forum for Information Retrieval Evaluation (FIRE) – 2010, Gandhinagar, India.

Pinaki Bhaskar, Somnath Banerjee, Snehasis Neogi, and Sivaji Bandyopadhyay. 2012a. *A Hybrid QA System with Focused IR and Automatic Summarization for INEX 2011*. In: Geva, S., Kamps, J., Schenkel, R.(eds.): Focused Retrieval of Content and Structure: 10th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2011. Lecture Notes in Computer Science, vol. 7424. Springer Verlag, Berlin, Heidelberg.

Pinaki Bhaskar, and Sivaji Bandyopadhyay. 2012b. *Answer Extraction of Comparative and Evaluative Question in Tourism Domain*. In: International Journal of Computer Science and Information Technologies (IJCSIT), ISSN: 0975-9646, Vol. 3, Issue 4, pp. 4610 – 4616.

Pinaki Bhaskar, and Sivaji Bandyopadhyay. 2012c. *Cross Lingual Query Dependent Snippet Genera-*

tion. In: International Journal of Computer Science and Information Technologies (IJCSIT), ISSN: 0975-9646, Vol. 3, Issue 4, pp. 4603 – 4609.

Pinaki Bhaskar, and Sivaji Bandyopadhyay. 2012d. *Language Independent Query Focused Snippet Generation*. In: T. Catarci et al. (Eds.): Information Access Evaluation. Multilinguality, Multimodality, and Visual Analytics: Third International Conference of the CLEF Initiative, CLEF 2012, Rome, Italy, Proceedings, Lecture Notes in Computer Science Volume 7488, pp 138-140, DOI 10.1007/978-3-642-33247-0_16, ISBN 978-3-642-33246-3, ISSN 0302-9743, Springer Verlag, Berlin, Heidelberg, Germany.

Pinaki Bhaskar, Somnath Banerjee, and Sivaji Bandyopadhyay. 2012e. *A Hybrid Tweet Contextualization System using IR and Summarization*. In: the Initiative for the Evaluation of XML Retrieval, INEX 2012 at Conference and Labs of the Evaluation Forum (CLEF) 2012, Pamela Forner, Jussi Karlgren, Christa Womser-Hacker (Eds.): CLEF 2012 Evaluation Labs and Workshop, pp. 164-175, ISBN 978-88-904810-3-1, ISSN 2038-4963, Rome, Italy.

Pinaki Bhaskar, Partha Pakray, Somnath Banerjee, Samadrita Banerjee, Sivaji Bandyopadhyay, and Alexander Gelbukh. 2012f. *Question Answering System for QA4MRE@CLEF 2012*. In: Question Answering for Machine Reading Evaluation (QA4MRE) at Conference and Labs of the Evaluation Forum (CLEF) 2012, Rome, Italy.

Pinaki Bhaskar, Bidhan Chandra Pal, and Sivaji Bandyopadhyay. 2012g. *Comparative & Evaluative QA System in Tourism Domain*. In: Meghanathan, N., Wozniak, M.(eds.): Computational Science, Engineering and Information Technology: the Second International Conference on Computational Science, Engineering and Informationa Technology (CCSEIT-2012), ACM International Conference Proceeding Series. ICPS, pp. 454-460, Coimbatore, India.

Pinaki Bhaskar, Kishorjit Nongmeikapam, and Sivaji Bandyopadhyay. 2012h. *Keyphrase Extraction in Scientific Articles: A Supervised Approach*. In: 24th International Conference on Computational Linguistics (Coling 2012), pp. 17-24, IIT, Bombay, Mumbai, India.

Pinaki Bhaskar 2013a. *A Query Focused Language Independent Multi-document Summarization*. Jian, A. (Eds.), ISBN 978-3-8484-0089-8, LAMBERT Academic Publishing, Saarbrücken, Germany.

Pinaki Bhaskar 2013b. *Multi-document Summarization using Automatic Key-phrase Extraction*. In: Student Research Workshop in the Recent Advances in Natural Language Processing (RANLP), Hissar, Bulgaria.

Pinaki Bhaskar, Somnath Banerjee, and Sivaji Bandyopadhyay. 2013c. *Tweet Contextualization (Answering Tweet Question) – the Role of Multi-document Summarization*. In: the Initiative for the Evaluation of XML Retrieval, INEX 2013 at CLEF 2013 Conference and Labs of the Evaluation Forum, Valencia, Spain.

Pinaki Bhaskar, Somnath Banerjee, Partha Pakray, Samadrita Banerjee, Sivaji Bandyopadhyay, and Alexander Gelbukh. 2013d. *A Hybrid Question Answering System for Multiple Choice Question (MCQ)*. In: Question Answering for Machine Reading Evaluation (QA4MRE) at CLEF 2013 Conference and Labs of the Evaluation Forum, Valencia, Spain.

Sibabrata Paladhi, and Sivaji Bandyopadhyay. 2008. *A Document Graph Based Query Focused Multi-Document Summarizer*. In: the 2nd International Workshop on Cross Lingual Information Access (CLIA), pp. 55-62.

Sivaji Bandyopadhyay, Amitava Das, and Pinaki Bhaskar. 2008. *English Bengali Ad-hoc Monolingual Information Retrieval Task Result at FIRE 2008*. In: the Forum for Information Retrieval Evaluation (FIRE) - 2008, Kolkata, India.

Somnath Banerjee, Partha Pakray, Pinaki Bhaskar, and Sivaji Bandyopadhyay, Alexander Gelbukh. 2013. *Multiple Choice Question (MCQ) Answering System for Entrance Examination*. In: Question Answering for Machine Reading Evaluation (QA4MRE) at CLEF 2013 Conference and Labs of the Evaluation Forum, Valencia, Spain.

Udo Hahn, and Martin Romacker. 2001. *The SYNDIKATE text Knowledge base generator*. In: the first International conference on Human language technology research, Association for Computational Linguistics , ACM, Morristown, NJ, USA.

Utsab Barman, Pintu Lohar, Pinaki Bhaskar, and Sivaji Bandyopadhyay. 2012. *Ad-hoc Information Retrieval focused on Wikipedia based Query Expansion and Entropy Based Ranking*. In: the Forum for Information Retrieval Evaluation (FIRE) – 2012, Kolkata, India.