Multi-Document Summarization using Automatic Key-Phrase Extraction

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

Abstract

The development of a multi-document summarizer using automatic key-phrase extraction has been described. This summarizer has two main parts; first part is automatic extraction of Key-phrases from the documents and second part is automatic generation of a multi-document summary based on the extracted key-phrases. The CRF based Automatic Key-phrase extraction system has been used here. A document graph-based topic/query focused automatic multi-document summarizer is used for summarization where extracted key-phrases are used as topic. The summarizer has been tested on the standard TAC 2008 test data sets of the Update Summarization Track. Evaluation using the ROUGE-1.5.5 tool has resulted in ROUGE-2 and ROUGE–SU-4 scores of 0.10548 and 0.13582 respectively.

1 Introduction

Text Summarization, as the process of identifying the most salient information in a document or set of documents (for multi document summarization) and conveying it in less space, became an active field of research in both Information Retrieval (IR) and Natural Language Processing (NLP) communities. Summarization shares some basic techniques with indexing as both are concerned with identification of the essence of a document. Also, high quality summarization requires sophisticated NLP techniques in order to deal with various Parts Of Speech (POS) taxonomy and inherent subjectivity. Typically, one may distinguish various types of summarizers.

Multi document summarization requires creating a short summary from a set of documents, which concentrate on the same topic. Sometimes an additional query is also given to specify the information need of the summary. Generally, an effective summary should be relevant, concise and fluent. It means that the summary should cover the most important concepts in the original document set, contains less redundant information and should be well organized.

In this paper, we proposes a multi-document summarizer, based on key-phrase extraction, clustering technique and sentence fusion. Unlike traditional extraction based summarizers, which do not take into consideration the inherent structure of the document, our system will add structure to documents in the form of graph. During initial preprocessing, text fragments are identified from the documents, which constitute the nodes of the graph. Edges are defined as the correlation measure between nodes of the graph. We define our text fragments as sentence.

First, during preprocessing stage it performs some document-based tasks like identifying seed summary nodes and constructing graph over them. Then key-phrase extraction module extracts the key-phrases form the documents and it performs key-phrase search over the cluster to find a sentence identifying relevant phrases. With the relevant phrases, the new compressed sentence has been constructed and then fused for summary. The performance of the system depends much on the identification of relevant phrases and compression of the sentences where the previous one again highly depends on the key-phrase extraction module.

Although, we have presented all the examples in the current discussion for English language only, we argue that our system can be adapted to work on other language (i.e. Hindi, Bengali etc.) with some minor addition in the system like incorporating language dependent stop word list, the stemmer and the parser for the language.

Proceedings of the Student Research Workshop associated with RANLP 2013, pages 22–29,
Hissar, Bulgaria, 9-11 September 2013.

2 Related Work

Currently, most successful multi-document summarization systems follow the extractive summarization framework. These systems first rank all the sentences in the original document set and then select the most salient sentences to compose summaries for a good coverage of the concepts. For the purpose of creating more concise and fluent summaries, some intensive post-processing approaches are also appended on the extracted sentences. For example, redundancy removal (Carbonell and Goldstein, 1998) and sentence compression (Knight and Marcu, 2000) approaches are used to make the summary more concise. Sentence re-ordering approaches (Barzilay et al., 2002) are used to make the summary more fluent. In most systems, these approaches are treated as independent steps. A sequential process is usually adopted in their implementation, applying the various approaches one after another.

A lot of research work has been done in the domain of multi-document summarization (both query dependent and independent). 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 proposed for generating query independent summaries. Websumm (Mani and Bloedorn, 2000) uses a graph-connectivity model to identify salient information. Zhang et al. (2004) proposed the methodology of correlated summarization for multiple news articles. In the domain of single document summarization a system for query-specific document summarization has been proposed (Varadarajan and Hristidis, 2006) based on the concept of document graph. A document graph-based query focused multi-document summarization system is described by Bhaskar and Bandyopadhyay, (2010a and 2010b). In the present work, the same summarization approach has been followed. As this summarizer is query independent, it extract the key-phrases and then the extracted key-phrases are used as query or keywords.

Works on identification of key-phrase using noun phrase are reported in (Barker and Cornacchia, 2000). Noun phrases are extracted from a text using a base noun phrase skimmer and an off-the-shelf online dictionary. Key-phrase Extraction Algorithm (KEA) was proposed in order to automatically extract key-phrase (Witten et al., 1999). The supervised learning methodologies have also been reported (Frank et al, 1999). Some works have been done for automatic keywords extraction using CRF technique. A comparative study on the performance of the six keyword extraction models, i.e., CRF, SVM, MLR, Logit, BaseLine1 and BaseLine2 has been reported in (Chengzhi et al., 2008). The study shows that CRF based system outperforms SVM based system. Bhaskar and Bandyopadhyay (2012) have developed a supervised system for automatic extraction of Key-phrases using Conditional Random Fields (CRF).

First a key-phrase extraction system has been developed based on the Bhaskar and Bandyopadhyay's (2012) method. Then a graph-based summarization system has been developed, where the key-phrase extraction system has been integrated for extraction key-phrases from document, which are serves as query or topic during summary generation.

3 Document-Based Process

3.1 Graph-Based Clustered Model

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

(1) The document set $D = \{d_1, d_2, \ldots d_n\}$ 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) At query time, each node is assigned a key-phrase dependent score and the search graph is expanded.

(5) A key-phrase 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 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.

3.2 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.

3.3 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 query independent part of the Markov clusters completes the document-based processing phase of the system.

4 Key-Phrase Extraction

A CRF based key-phrase extraction system as described by Bhaskar et al. (2012) is used to extract key-phrases from the documents.

4.1 Features Identification for the System

Selection of features is important in CRF. Features used in the system are,

F = {Dependency, POS tag(s), Chunk, NE, TF, Title, Body, Stem of word, W_{i-m}, …, W_{i-1}, W_i, W_{i+1},… , W_{i-n} }.

The features are detailed as follows:

i) **Dependency parsing:** Some of the key-phrases are multiword. So relationship of verb with subject or object is to be identified through dependency parsing and thus used as a feature.

ii) **POS feature:** The Part of Speech (POS) tags of the preceding word, the current word and the following word are used as a feature in order to know the POS combination of a key-phrase.

iii) **Chunking:** Chunking is done to mark the Noun phrases and the Verb phrases since much of the key-phrases are noun phrases.

iv) **Named Entity (NE):** The Named Entity (NE) tag of the preceding word, the current word and the following word are used as a feature in order to know the named entity combination of a key-phrase.

v) **Term frequency (TF) range:** The maximum value of the term frequency (max_TF) is divided into five equal sizes (*size_of_range)* and each of the term frequency values is mapped to the appropriate range (0 to 4). The term frequency range value is used as a feature. i.e.

$$size\_of\_range = \frac{\max\_TF}{5} \quad (2)$$

Thus Table 1 shows the range representation. This is done to have uniform values for the term frequency feature instead of random and scattered values.

Class	Range
0 to size_of_range	*0*
*size_of_range + 1 to 2*size_of_range*	*1*
*2*size_of_range + 1 to 3*size_of_range*	*2*
*3*size_of_range + 1 to 4*size_of_range*	*3*
*4*size_of_range + 1 to 5*size_of_range*	*4*

Table 1: Term frequency (TF) range

vi) **Word in Title:** Every word is marked with T if found in the title else O to mark other. The title word feature is useful because the words in title have a high chance to be a key-phrase.

vii) **Word in Body:** Every word is marked with B if found in the body of the text else O to mark other. It is a useful feature because words present in the body of the text are distinguished from other words in the document.

viii) **Stemming:** The Porter Stemmer algorithm is used to stem every word and the output stem for each word is used as a feature. This is because words in key-phrases can appear in different inflected forms.

ix) **Context word feature:** The preceding and the following word of the current word are considered as context feature since key-phrases can be a group of words.

4.2 Generating Feature File for CRF

The features used in the key-phrase extraction system are identified in the following ways.

Step 1: The dependency parsing is done by the Stanford Parser[1]. The output of the parser is modified by making the word and the associated tags for every word appearing in a line.

Step 2: The same output is used for chunking and for every word it identifies whether the word is a part of a noun phrase or a verb phrase.

Step 3: The Stanford POS Tagger[2] is used for POS tagging of the documents.

Step 4: The term frequency (*TF*) range is identified as defined before.

Step 5: Using the algorithms described by Bhaskar et al. (2012), every word is marked as *T* or *O* for the title word feature and marked as *B* or *O* for the body word feature.

Step 6: The Porter Stemming Algorithm[3] is used to identify the stem of every word that is used as another feature.

Step 7: In the training data with the combined key-phrases, the words that begin a key-phrase are marked with *B-KP* and words that are present intermediate in a key-phrase are marked as *I-KP*. All other words are marked as *O*. But for test data only *O* is marked in this column.

4.3 Training the CRF and Extracting Key-Phrases

A template file was created in order to train the system using the feature file generated. After training the C++ based CRF++ 0.53 package[4], a model file is produced. The model file is required to run the system. The feature file is again created from the document set. After running this files into the system, the system produce the output file with the key-phrases marked with *B-KP* and *I-KP*. All the Key-phrases are extracted from the output file and stemmed using Porter Stemmer. Now, these extracted key-phrases are used as query or topic to generate the summary.

5 Key-Phrase Dependent Process

After key-phrase extraction, first the nodes of the already constructed search graph are given a key-phrase dependent score. With the combined scores of key-phrase independent score and key-phrase 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.

5.1 Recalculate the Cluster Score

There are two basic components in the sentence weight like key-phrases dependent scores and synonyms of key-phrases dependent scores. We collect the list of synonyms of the each word in the key-phrases from the WordNet 3.0[5]. The term dependent score (both for key-phrases and synonyms) are calculated using equation 2.

$$w = \sum_{t=1}^{n_t}(n_t - t + 1)\left(\sum_p\left(1 - \frac{f_p^{t-1}}{N_i}\right)\right) \times b \quad (3)$$

where, w is the term dependent score of the sentence i, t is the no. of the term, n_t is the total

[1] http://nlp.stanford.edu/software/lex-parser.shtml
[2] http://nlp.stanford.edu/software/tagger.shtml
[3] http://tartarus.org/~martin/PorterStemmer/
[4] http://crfpp.sourceforge.net/
[5] http://wordnet.princeton.edu/

no. of term, f_p^t is the possession of the word which was matched with the term t in the sentence i, N_s is the total no. of words in sentence i and b is boost factor of the term, which is 2 or 1 for key-phrases and synonyms respectively. These two components are added to get the final weight of a sentence.

5.2 Recalculate the Cluster Ranking

We start by defining a function that attributes values to the sentences as well as to the clusters. We refer to sentences indexed by i and key-phrases indexed by l. We want to maximize the number of key-phrase covered by a selection of sentences:

$$maximize \sum_l w_l^k k_l \quad (4)$$

where, w_l^k is the weight of key-phrase l in the sentence i and k_l is a binary variable indicating the presence of that key-phrase in the cluster.

We also take the selection over the synonyms of the key-phrases. The general sets of synonyms are indexed by s. So we also want to maximize the number of synonyms covered by a selection of sentences using similar calculation like for key-phrase using equation 2.

So, the key-phrase dependent score of a cluster is the weighted sum of the key-phrases it contains. If clusters are indexed by x, the key-phrase dependent score of the cluster x is:

$$c_x^k = \sum_{i=x_1}^{x_n} \sum_l w_l^k k_l + \sum_{i=x_1}^{x_n} \sum_s w_s^s k_s \quad (5)$$

where, c_x^k is the key-phrase dependent score of the cluster x, x_1 is the starting sentence number and x_n is the ending sentence number of the cluster x. Now, the new recalculated combined score of cluster x is:

$$c_x = c_x^g + c_x^k \quad (6)$$

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

5.3 Retrieve Sentences for Summary

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

$$\max\left(\sum_l w_l^k k_l + \sum_s w_s^s k_s\right) \forall i, x_1 \le i \le x_n \quad (7)$$

where, x_1 is the first sentence and x_n is the n^{th} i.e. last sentence of a cluster.

The highest weighted two sentences are taken from each cluster in order one by one. 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[6] is used to parse the sentences and get the phrases of the sentence.

5.4 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 key-phrase, 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 key-phrase 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.

5.5 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:

$$\sum_i l_i S_i < L \quad (8)$$

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.

6 Sentence Ordering and Coherency

In this paper, we will propose a scheme of ordering which is different from the above two approaches 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 sum-

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

mary 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 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 $(p\text{-}1)^{th}$ sentence.

7 Evaluation

We evaluate our summaries by ROUGE[7], an automatic evaluation tool. We have run our system on Text Analysis Conference (TAC, formerly DUC, conducted by NIST) 2008 Update Summarization track's data sets[8]. This data set contains 48 sets and each set has two subsets of 10 documents, i.e. there are 960 documents. The evaluation data set has 4 model summaries for each document set, i.e. 8 model summaries for each set. We have evaluated our output summaries on those model summaries using ROUGE-1.5.5. The baseline scores provided by the organizer were 0.058 and 0.093 of ROUGE-2 and ROUGE-SU4 respectively. The system's score is 0.10548 and 0.13582 respectively. All the results are shown in table 2. The comparison of ROUGE-2 and ROUGE-SU4 among the proposed system, the system developed by Bhaskar and Bandyopadhyay (2010b), the best system of TAC 2008 Update Summarization track and the baseline system of TAC 2008 Update Summarization track are also shown in table 2.

8 Conclusion and Future Work

In this work we present a graph-based approach for multi document summarization system using automatic key-phrase extraction. The experimental results suggest that our algorithm is effective. It can be used in web based system like search engine or QA system, where offline summary of multiple document on same topic can be pre-generated and will be used during online phase, which will reduce many burden on online modules. The proposed algorithm can be improved to handle more noisy WEB articles or work on other domain too.

As the topic or query are given to the system along with the document sets, it's has been extracted automatically as key-phrases. The key-phrase extraction module is not 100% accurate and sometimes extracts some extra or noisy phrases as key-phrase. Hence the performance of the summarizer slightly decreases. But it is very useful where the topic or query is not available and we still need the summary from documents.

The important aspect is that our system can be tuned to generate summary with custom size specified by users. Lastly, it is shown that our system can generate summary for other non-English documents also if some simple resources of the language like stemmer and parser are available.

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

ROUGE Evaluation	**Average_R**				Average_P		Average_F	
	Proposed System	Bhaskar et al. (2010b)'s System	Top score of TAC 2008	Baseline of TAC 2008	**Proposed System**	Bhaskar et al. (2010b)'s System	**Proposed System**	Bhaskar et al. (2010b)'s System
ROUGE-1	0.50626	0.53291	-	-	0.48655	0.51216	0.49512	0.52118
ROUGE-2	**0.10548**	**0.11103**	**0.111**	**0.058**	**0.09248**	**0.09735**	**0.09491**	**0.09991**
ROUGE-3	0.03301	0.03475	-	-	0.03061	0.03223	0.03169	0.03336
ROUGE-4	0.01524	0.01604	-	-	0.01397	0.01471	0.01454	0.01530
ROUGE-L	0.37204	0.39162	-	-	0.35727	0.37607	0.36368	0.38282
ROUGE-W-1.2	0.12407	0.13060	-	-	0.21860	0.23011	0.16027	0.16870
ROUGE- SU4	**0.13582**	**0.14297**	**0.143**	**0.093**	**0.12693**	**0.13361**	**0.12954**	**0.13636**

Table 2: Evaluation scores of ROUGE

[7] http://berouge.com/default.aspx
[8] http://www.nist.gov/tac/data/index.html

References

Chengzhi ZHANG, Huilin WANG, Yao LIU, Dan WU, Yi LIAO, and Bo WANG. 2008. *Automatic keyword Extraction from Documents Using Conditional Random Fields*. Journal of Computational Information Systems, 4:3, pp. 1169-1180.

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

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

Ernesto D'Avanzo, and Bernardo Magnini. 2005. *A Keyphrase-Based Approach to Summarization:the LAKE System at DUC-2005*. In: Proc. of Document Understanding Conferences.

Eibe Frank, Gordon Paynter, Ian Witten, Carl Gutwin, and Craig Nevill-Manning. 1999. *Domain-specific keyphrase extraction*. In: the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99), pp. 668-673, California.

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.

Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. *KEA:Practical Automtic Key phrase Extraction.* In: the fourth ACM conference on Digital libraries, pp. 254-256, ISBN:1-58113-145-3, ACM New York, NY, USA.

Inderjeet Mani, and Eric Bloedorn. 1999. *Summarizing Similarities and Differences Among Related Documents*. In: Information Retrieval, Volume 1, Issue 1-2, pp. 35-67, Kluwer Academic Publishers Hingham, MA, USA.

Jaime Carbonell, and Jade Goldstein. 1998. *The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries*. In: the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ISBN:1-58113-015-5, pp. 335-336, NY, USA.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.* In: Proc. of the 18th International Conference on Machine Learning (ICML01), pp. 282-289, ISBN: 1-55860-778-1, Williamstown, MA, USA.

Ken Barker, and Nadia Cornacchia. 2000. *Using noun phrase heads to extract document keyphrases*. In: Proc. of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence. Canada. pp. 40-52.

Kevin Knight, and Daniel Marcu. 2000. *Statistics-based summarization --- step one: Sentence compression*. In: the American Association for Artificial Intelligence Conference (AAAI-2000), pp. 703--710.

Martin Porter. 1980. *An algorithm for suffix stripping*. Program, 14(3), 130–137.

Peter Turney. 1999. *Learning to Extract Keyphrases from Text*. National Research Council, Institute for Information Technology, Technical Report ERB-1057. (NRC \#41622).

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, 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. *Cross Lingual Query Dependent Snippet Generation*. 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. 2012c. *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. 2012d. *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, Kishorjit Nongmeikapam, and Sivaji Bandyopadhyay. 2012e. *Keyphrase Extraction in Scientific Articles: A Supervised Approach*. In: the proceedings of 24th International Conference on Computational Linguastics (Coling 2012), pp. 17-24, 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. *Answering Questions from Multiple Documents – the Role of Multi-document Summarization*. 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.

Ramakrishna Varadarajan, and Vagelis Hristidis. 2006. *A system for query specific document summarization*. In: the 15th ACM international conference on Information and knowledge management, pp. 622-631, ISBN: 1-59593-433-2, ACM New York, NY, USA.

Regina Barzilay, Noemie Elhadad, and Kathleen R. McKeown. 2002. *Inferring strategies for sentence ordering in multidocument news summarization*, In: Artificial Intelligence Research. 17, pp. 35—55

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

Stijn Van Dongen. 2000a. *A stochastic uncoupling process for graphs*. Report No. INS- R0011, Centre for Mathematics and Computer Science(CWI), Amsterdam.

Stijn Van Dongen. 2000b. *Graph clustering by flow simulation*. PhD Thesis, University of Utrecht, The Netherlands.

Su Nam Kim and Min-Yen Kan. 2009. *Re-examining Automatic Keyphrase Extraction Approaches in Scientific Articles*. In: Proc. of the 2009 Workshop on multiword Expressions, ACL-IJCNLP 2009. Suntec, Singapore. pp. 9-16.

Ya Zhang, Chao-Hsien Chu, Xiang Ji, and Hongyuan Zha. 2004. *Correlating Summarization of Multi-Source News with K-Way Graph Biclustering*. In: SIGKDD Explorations. 6(2), Association for Computing Machinery. Volume 6 Issue 2, pp. 34-42, ACM New York, NY, USA.