# A Survey of Unstructured Text Summarization Techniques

Sherif Elfayoumy
School of Computing
University of North Florida
Jacksonville, Florida

Jenny Thoppil
School of Computing
University of North Florida
Jacksonville, Florida

***Abstract*—Due to the explosive amounts of text data being created and organizations increased desire to leverage their data corpora, especially with the availability of Big Data platforms, there is not usually enough time to read and understand each document and make decisions based on document contents. Hence, there is a great demand for summarizing text documents to provide a representative substitute for the original documents. By improving summarizing techniques, precision of document retrieval through search queries against summarized documents is expected to improve in comparison to querying against the full spectrum of original documents.**

**Several generic text summarization algorithms have been developed, each with its own advantages and disadvantages. For example, some algorithms are particularly good for summarizing short documents but not for long ones. Others perform well in identifying and summarizing single-topic documents but their precision degrades sharply with multi-topic documents. In this article we present a survey of the literature in text summarization. We also surveyed some of the most common evaluation methods for the quality of automated text summarization techniques. Last, we identified some of the challenging problems that are still open, in particular the need for a universal approach that yields good results for mixed types of documents.**

***Keywords—text summarization; unstructured data; text mining; unstructured data analytics***

## I. Introduction

The rapid growth of online information services, social media and other digital format documents means that huge amounts of information are becoming immediately available and readily accessible to a large number of end-users. However, human ability to organize and understand a large number of documents is limited. This well-known information overload problem is most acute when we need to make a decision or understand something deeply, which typically involves reviewing several documents, but have limited time. Reading through long documents consumes precious time in understanding the gist of the document.

Web search engines look for documents from the Internet based upon user supplied queries. They not only overwhelm users with too many results, they also provide documents that may not be very relevant to the topic being studied by the user. For example, if the user is searching using some keyword and the search engine finds it somewhere inside a document, that document will be a “search hit” even if the document is not really relevant to the keyword. The most common search method is based on maintaining an inverted list (text index) of documents’ text. Not only precision is hurt by indexing every word in the document, excluding stop words, but also efficiency is adversely impacted. If summaries are indexed and searched instead, index size will be considerably smaller and search hits will be of better quality (fewer false positives) [1]. This can be explained using the definition of Precision and Recall measures used in information retrieval. Precision is defined as the percentage of the relevant items in the returned set and Recall is the percentage of the relevant items in the returned set compared to those in the collection. If the whole collection is retrieved, then the Recall is 100%, but Precision is low. Most search engines suffer from this problem (high Recall and low Precision). If search engines search only a document’s primary ideas, instead of every word, then Recall will likely not be decreased but Precision will likely improve. Hence, an automated facility for summarizing documents to improve productivity is desirable. A good summarization system should include only sentences that are most important to a document’s theme; it must also cover all documents’ topics [2].

Using a summary instead of the whole document as a representative of what the document is about would mean processing a fraction (20% or less) of the document’s text, yet yield better Precision and lesser processing time for search queries. In order to determine the requirements of a good summarization system, many text summarization approaches were reviewed. An in-depth review of text summarization literature was conducted and results from this study along with a description of each algorithm, its strengths and weaknesses are presented in this article. Section II presents an overview of the major types of text summarization techniques. Section III provides detailed information on unsupervised text summarization techniques. The evaluation techniques used for assessing the quality of text summarization systems are also discussed in section IV. It was found that due to the shortcomings of the text summarization approaches currently available, there is a lack of a universal approach for document summarization that provides high Precision and Recall with various types of text corpora.

## II. Text Summarization by Classification

Many research papers and books related to natural language processing and computational linguistics were thoroughly investigated in order to determine current techniques used for automated text summarization and in particular their advantages and disadvantages. Text summarization techniques were classified by Hahn and Mani [3] as follows:

### *A. Query-relevant Summarization*

A query-relevant summary presents the document's contents that are closely related to an initial search query. This can be achieved by extending conventional information retrieval technologies. Depending on the user's supplied query, the text documents are searched for matches with that query, and a summary is created on the fly, which contains the sentences that have the query matches.

The selection of sentences based on their ranking, with respect to a query, using latent semantic analysis (LSA) was proposed by Gong and Liu [2]. Park et al. proposed a new approach using a combination of Non-negative Matrix Factorization and K-means clustering to identify sentences based on a query. Their approach produced better performance than LSA [4]. Tang et al. retrieve relevant documents to a query, use a unified probabilistic approach to discover query-oriented topics and apply four scoring methods to calculate the importance of each sentence. Sentences with the highest score make the summary of each document [5].

### *B. Generic Summarization*

A generic summary provides an overall sense of the document's contents. It contains the main topics of the document, while keeping redundancy to a minimum. As neither query nor topic is provided to the summarization process, it is challenging to develop a high quality generic summarization method [2]. Generally, text summary extraction from a document can be done using one or more of the following approaches:

#### *a) Sentence Extraction*

In this method, original pieces from the source document are selected and concatenated to yield a shorter text. This technique is easy to adapt to large sources of data. A Conditional Random Field (CRF) framework was proposed by Shen et al. In their framework, the summarization problem is viewed as a sequence labeling problem where a document is a sequence of sentences that are labeled as 1 or 0 based on the label assignment to other sentences [6]. Daume and Marcu presented BAYESUM which is a Bayesian Summarization model for query expansion. This model was found to be work well in purely extractive settings [7].

#### *b) Sentence Abstraction*

This method paraphrases in more general terms what the text is about. This is done using very sophisticated algorithms. It is easy to adapt to higher compression rates [3]. Kinght and Marcu presented corpus-based methods for attacking the sentence abstraction problem, one using the noisy-channel framework, and other using a decision-based model. While most corpus-based work focuses on keyword extraction, this work focused on constructing new whole sentences by analyzing existing, manually produced, compressions [8].

#### *c) Supervised Approaches*

These approaches make use of human-made summaries or extracts to identify features or parameters of summarization algorithms. In these methods, a human user decides which parameters are important for text summary and accordingly the summary is generated. Bravo-Marquez and Manriquez trained ranking functions using linear regressions and ranking SVMs, which are also combined using Borda count [9]. Top ranked sentences are concatenated and used to build summaries, which are compared with the first sentences of the distant summary using ROUGE evaluation measures [10]. Experimental results obtained showed that the combination of different ranking techniques improves the quality of the generated summary.

#### *d) Unsupervised Approaches*

These approaches determine the relevant parameters without regard to human–made summaries [11]. The summary is generated without any user input. Probabilistic Latent Semantic Indexing (PLSI) is an unsupervised learning method based on statistical latent class models. PLSI was applied to document clustering by Hoffman [12]. In contrast to standard

Latent Semantic Indexing (LSI) by Singular Value Decomposition, the probabilistic variant, PLSI, has a solid statistical foundation and defines a proper generative data model. Retrieval experiments indicated substantial performance gains over LSI. PLSI was further developed into a more comprehensive Latent Dirichlet Allocation (LDA) model by Blei et al [13]. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. Topic probabilities provide an explicit representation of a document.

The unsupervised approaches do not require user input in deciding the important features of the document, requiring a more sophisticated algorithm to compensate for lack of human intervention. We believe unsupervised summaries provide a higher level of automation which makes them more suitable for processing Big Data.

## III. UNSUPERVISED GENERIC TEXT SUMMARIZATION

In this section we investigate further generic text summarization using unsupervised approaches for sentence extraction. Generic text summary can improve the processing time and precision of information retrieval, since the summary has to be created only once and contains the most important themes of the document. In contrast, query-related summaries need to be created every time a query is provided by the user. Moreover, it is possible the summaries do not have the query as the main topic of the documents retrieved. The sentence extraction approach is a simpler but effective way of extracting main themes of the document as compared to sentence abstraction, which involves many complicated linguistic and natural language processing algorithms that require a lot of processing time. The following generic unsupervised text summarization algorithms have been amongst the most prominent in the literature.

### *A. Cosine Similarity*

The vector space model using cosine measure is one of the most widely used models for text retrieval, mainly because of its conceptual simplicity. Sentences and queries are represented in a high-dimensional space, in which each dimension of the space corresponds to a word in a sentence collection [14]. The most relevant sentences for a query are expected to be those represented by the vectors closest to the query.

This method can be slightly modified to calculate a weight for each sentence with respect to its relevance to the entire document. In order to calculate the cosine measure of a sentence, the frequency of each term in the entire document (*docfreq*) and the frequency of the term in a particular sentence *(termfreq*) are calculated. Then for each sentence, i.e. query, the cosine angle between the query and the entire document is calculated using the formula below. If the cosine measure is highest, i.e. the cosine angle between query and document is smallest, then that sentence is the most relevant to the document. Thus the sentences are ranked according to their cosine measures and a summary is created using top ranked sentences. The formula for cosine similarity is as follows:

$$\cos(\mathrm{termfreq,docfreq}) = \frac{\sum_{i=1}^{n} termfreq.docfreq}{\left(\sqrt{\sum_{i=1}^{n} termfreq^2} \times \sqrt{\sum_{i=1}^{n} docfreq^2}\right)}$$

where *n* = number of terms per sentence

The Cosine Similarity technique is not well-suited for obtaining diverse topics in a document, although it does an excellent job of selecting the most relevant sentences in the document. In Maximal Marginal Relevance (MMR), the Cosine Similarity technique is changed to add diversity to the document summary [15].

### B. Relevance Measure

Gong and Liu [2] proposed a relevance measure algorithm, which is also based on ranking sentences using their relevance scores. This algorithm works as follows: The weighted frequency vector is obtained for each sentence using the local weight of each term and its global weight over the document, where each term's weight is obtained as

$$aji = L(tji).G(tji)$$

where *L(tji)* is the local weight for term j in passage i and *G(tji)* is the global weight for term j.

Vector length normalization, also referred to as cosine normalization, is carried out and the weight of each sentence is obtained. The sentence with highest relevance score is extracted and added it to summary. All the terms contained in the sentence are deleted from the original document. The sentence itself is deleted and weighted term frequency vector for the document is recomputed. Again sentence with highest relevant score is found and this process is continued until the number of sentences in the summary reaches a predefined value [2].

### C. Latent Semantic Analysis using SVD

Singular value decomposition (SVD) is a method of word co-occurrence analysis using a dimensionality reduction approach. In the process of dimensional reduction, co-occurring terms are mapped onto the same dimensions in the reduced space, thus increasing similarity in the representation of semantically similar sentences [15]. In this method, the weight of the sentences is first obtained using the same principle as described in [2] and then a sentence matrix A= [A1, A2, …An] with each column vector Ai representing the weighted term vector of sentence i is created. If there are m terms and n sentences, then matrix A is of dimension m*n. Using singular value decomposition, A= USV$^T$, where the columns of U (m*dimension) are left singular vectors, S (dimension * dimension) gives the non-negative singular values, and V$^T$ (dimension *n) columns are right singular vectors. The first right singular vector is selected and the sentence with the largest index value is selected and included in the summary [2]. The next right singular vector representing the next dimension is selected and the largest index valued sentence is added to the summary. Thus, this method chooses sentences from every dimension covering all topics in the document.

In Enhanced Latent Semantic Analysis using SVD [16], for each sentence vector in matrix V, its components are multiplied by corresponding singular values, to compute each sentence length. The reason for using the multiplication is to favor the index values in the matrix V that correspond to the highest singular values; i.e. the most significant topics. The sentence weight is calculated as follows:

$$S_k = \sqrt{\sum_{i=1}^{n} V_{k,i}^2 \times S_i^2}$$

where $S_k$ is the sentence with sentence number k and n = number of dimensions.

The Latent Semantic Analysis using SVD, though a good dimensionality reduction technique, has two disadvantages. It is necessary to use the same number of dimensions as the number of sentences chosen for a summary. If a high number of dimensions of the reduced space is chosen, the probability of selecting a significant topic in the summary is reduced. Hence, it may not give the most relevant sentences for longer documents. Also, sentences with large sentence weights, but not the largest (they do not win in any dimension), will not be chosen although contents may be very suitable for the summary [16]. Hence in Enhanced Latent Semantic Analysis technique, the weight of each sentence is further calculated with respect to the entire document, not just with respect to each dimension, so that sentences can be correctly ranked.

### D. Maximal Marginal Relevance (MMR)

MMR is based on the vector space model of text retrieval [15][17] and is well suited for query-based and multi-document summarization. It chooses sentences according to a weighted combination of their relevance to a query and their redundancy with sentences that have already been extracted using Cosine Similarity. The MMR score $S_{CMMR(i)}$ for a given sentence $S_i$ in a document is given by

$$S_{CMMR(i)} = [\lambda Sim(S_i, D) - (1-\lambda)\max(S_i, Summ)]$$

where *D* is the average document vector, *Summ* is the average vector from the set of sentences already selected, and λ trades off between relevance and redundancy. *Sim* is the cosine similarity between the two documents.

When λ=1, it computes the incrementally standard relevance ranked list. When λ=0, it computes a maximal diversity ranking among the documents. When MMR was compared with Enhanced LSA, MMR yielded better Precision [17]. The Maximal Marginal Relevance measure is commonly used for multi-document summarization.

### *E. Full Coverage Summarizer*

The first phase in the Full Coverage Algorithm is to parse a document into sentences [18]. During this phase, stop-words are removed and the Porter stemming algorithm is applied to stem the words in the document to their base forms. The entire document is then treated as a query to each individual sentence. The second step is to calculate the subset of sentences that cover the entire concept space of the document. The highest ranked sentence is selected using Cosine Similarity. The words that appear in the highest ranked sentence are removed from the query and the process is repeated until no words can be removed from the query, thus obtaining the summarized document. Mallett et al. also compared the Full-Coverage summarizer with MEAD and found that the Full-Coverage summarizer outperforms the MEAD clustering technique [18].

### *F. MEAD*

MEAD is a multi-document summarizer which generates summaries using cluster centroids produced by topic detection and tracking system (TDT) [19]. MEAD uses the online document clustering system, CIDR, to produce the clusters and then uses its own weighting scheme to rank the sentences in the cluster. The CIDR algorithm initially places the first document by itself in the first cluster. The centroids of the cluster are a group of words that represent a cluster of documents. When new sentences are processed, they are compared with the centroids of the existing cluster. Centroids of a cluster are the weighted averages of the *tf*idf* values of the documents already assigned in the cluster, where *tf = frequency of term* and *idf = inverse document frequency*.

Similarity between a document and a centroid is measured using the cosine (normalized inner product) of the corresponding *tf*idf* vector. If the similarity goes below a predefined threshold value, a new cluster is created.

Centroid-based summarization (CBS) uses the centroids of the clusters produced by CIDR to identify sentences central to the topic of the entire cluster. MEAD combines the following three parameters to find the score of a sentence within each cluster:

*1) Centroid value – The centroid value of sentence Si is computed as the sum of the centroid values Cw,i of all the words in the sentence.*

$$Ci = \sum C_{w,\,i}$$

*2) Positional Value – The first sentence in a document gets the same score Cmax as the highest-ranking sentence in the document using the centroid value. The score for all the sentences within the document is computed as:*

$$P_i = (n - i + 1) / n \times C_{\max}$$

*3) First sentence overlap – Overlap value is computed as the inner product of the sentence vectors for the current sentence i and the first sentence of the document.*

$$F_i = S_1 \times S_i$$

*4) Redundancy Penalty -*

$$Rs = 2 \times \left( \frac{noofoverlappingwords}{noofwords_{sentence1} \times noofwords_{sentence2}} \right)$$

$$Score(S_i) = w_c C_i + w_p P_i + w_f F_i + w_r R_s$$

Using this score, the sentences are ranked and chosen from each cluster in MEAD.

### *G. K-means Clustering Followed by tf.idf*

A modified K-means algorithm using the Minimum Description Length Principle (MDL) is used, where the number of clusters are estimated, which otherwise has to be supplied by the user [20]. Using K-means, the diversity in the document is obtained in the form of clusters. After clusters are identified, sentences in each cluster are ranked based on the *tf*idf* value, where *tf = term frequency* of each term and *idf = inverse document frequency*, using term frequency over the entire document (*doc*) and the weight of each sentence:

$$W_s = \sum\nolimits_{x=1}^{n} \left(1 + \log(tf(x).idf(x))\right)$$

where $n$ = number of terms per sentence,

*idf(x)= log (N/ doc(x))* where *N* = number of sentences

The weighting scheme is obtained to reduce the redundancy in the document and to choose the sentence with largest weight in the summary. Thus, one or more sentences are chosen from each cluster and added into the summary.

After reviewing the above algorithms, it was clear that each works well given some assumptions, but they do not fulfill all requirements in all circumstances. For example, Cosine Similarity is a good and simple algorithm, if the same words are used for explaining a certain situation. In such cases, it will give very good results. But if the same words are not repeated in the document for a particular context, its Precision is much reduced.

Enhanced Latent semantic analysis using SVD does a good job in finding co-occurrence of terms in a document. It is, therefore, able to find diverse topic areas in the document, but as the number of sentences in the document increases, its Precision drastically degrades, since the number of dimensions in the vector space increases. MMR is a good multi-topic summarizer, but it is not very effective for single-topic documents. Clustering techniques, MEAD, and K-means Clustering are time consuming.

## IV. Text Summarization Evaluation Techniques

Objectively evaluating the quality of summarizers is not an easy task, because there are various evaluation metrics. Moreover, arguably there is no “ideal” summary to compare against [21]. Typically, the base-line is a summary generated by a human being. The commonly used metrics include Precision, Recall, Kappa, Relative Utility and n-grams. They are used to compare the automated summary against the manually produced summary.

### A. Precision and Recall

Using Precision and Recall measures may be the simplest, but most effective evaluation technique used in text summarization. Precision is defined as the percentage of relevant sentences in the returned set and Recall is the percentage of the relevant sentences in the collection that are in the returned set [14]. $Sum_{manual} \cap Sum_{automated}$ is the set of sentences selected by both automated summarizer and manual summarizer where $Sum_{manual}$ is the set of sentences selected by manual summarizer and $Sum_{automated}$ are the sentences selected by the automated summarizer. Then Precision and Recall are calculated as follows:

$$\mathrm{Pr}\,ecision = \frac{Sum_{manual} \cap Sum_{automated}}{Sum_{automated}}$$

$$\mathrm{Re}\,call = \frac{Sum_{manual} \cap Sum_{automated}}{Sum_{manual}}$$

Normally there is more than one judge for summarizing a document manually, and the common sentences among the judges need to be taken as relevant sentences. The amount of agreement between the manual and automated summaries is an important factor in calculating Precision and Recall metrics. A drawback of using Precision and Recall only for evaluating summarizers is that agreement may be by chance and the Precision and Recall approach does not take chance agreement into account [21].

### B. Kappa Coefficient

Kappa is an evaluation measure which is increasingly used in NLP (Natural Language Processing) research. It factors out random agreement that Precision and Recall measures do not. Random agreement is defined as the level of agreement which would be reached by random annotation using the same distribution of categories as real annotators [21]. The Kappa coefficient (K) measures pair wise agreement among a set of judges making category judgments and is computed as follows:

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

where *P(A)* is the probability that the judges agree and *P(E)* is the probability of which judges are expected to agree by chance [22].

Using the Kappa Coefficient along with Precision and Recall gives an accurate evaluation of how well an automated summarizer performs compared to a manual summarizer.

### C. Relative Utility

Relative Utility (RU) is a measure for evaluating extractive summarizers. RU is applicable in both single-document and multi-document summarization. When the target sentences are given, the judges (manual and automated summarizers) pick different sentences. This is called Summary Sentence Substitutability (SSS) [23].

RU agreement is defined as the relative score that one judge would get, given his own extract and the other judge's sentence judgments. In RU, a number of judges are asked to assign utility scores to all n sentences in a document.

The top e sentences according to utility score are then used as a sentence extract of size e.

In situations where automated summaries are compared to manual summaries where sentences are not ranked, the Relative Utility technique could not be used as an evaluation technique.

### D. BLEU and n-grams

The main idea of the BLEU (Bilingual Evaluation Understudy) method is to measure the translation closeness between a candidate machine translation and a set of reference human translations with a numerical metric. In the unigram precision model, the precision is calculated by simply counting the number of candidate translation words (unigrams) which occur in any reference translation and then divide by the total number of words in the candidate translation [24]. Machine translation system can over-generate reasonable words; hence, the modified unigram technique first counts the maximum number of times a word occurs in single reference translation. Then the total count of each candidate word is clipped by its maximum reference count, the clipped counts are added and then divided by the total (unclipped) number of candidate words. The modified n-gram precision is computed similarly for any n.

The formula for modified n-gram precision on a block of text is as follows:

$$Pn = \frac{\sum_{C \in \{Candidates\}} \sum_{n-gram \in C} Count_{clip}(n-gram)}{\sum_{C \in \{Candidates\}} \sum_{n-gram` \in C`} Count(n-gram`)}$$

The BLEU technique is applicable only in situations where automated machine translations are performed.

## V. Conclusion

The document summarization problem is an interesting problem due to its impact on information retrieval methods as well as on the efficiency of decision making processes, particularly in the age of Big Data. Although a wide variety of text summarization techniques and algorithms have been developed there is a need for new approaches to produce precise and reliable document summaries that can tolerate differences in document characteristics.

We plan to use the best of bread among the existing techniques to create an ensemble that is capable of producing superior results on mixed document corpora.

## References

[1] Walters, William H. "Comparative Recall and Precision of Simple and Expert Searches in Google Scholar and Eight Other Databases," portal: Libraries and the Academy, vol. 11, no. 4, 2011, pp. 971-1006.

[2] Gong, Y. and X. Liu, "Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis," Proceedings of the 24th annual international ACM/SIGIR conference on research and development, ACM-Press, New Orleans, LA, 2001, pp. 75-95.

[3] Hahn, U. and I. Mani, "The Challenges of Automatic Summarization," IEEE Computer, vol. 33, no. 11, 2000, pp. 29-36.

[4] S. Park, J. Lee, D. Kim, and C. Ahn, "Multi-document Summarization Based on Cluster Using Non-negative Matrix Factorization," Lecture Notes in Computer Science, vol. 4362, pp. 761-770, 2007.

[5] J. Tang, L. Yao, and D. Chen, “Multi-topic Based Query-oriented Summarization,” Proceedings of SIAM International Conference on Data Mining (SDM), Sparks, NV, 2009.

[6] D. Shen, J. Sun, H. Li, Q. Yang, and Z. Chen, “Document Summarization Using Conditional Random Fields,” Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), 2007, pp. 2862–2867.

[7] H. Daumé and D. Marcu, “Bayesian Query-focused Summarization,” Proceedings of the 21st International Conference on Computational Linguistics, 2006, pp. 305-3012.

[8] K. Knight, and D. Marcu, “Summarization Beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression,” Artificial Intelligence Journal, vol. 139, no. 1, 2002, pp. 91–107.

[9] Bravo-Marquez, Felipe and Manriquez, Manuel, “A Zipf-Like Distant Supervision Approach for Multi-document Summarization Using Wikinews Articles,” Lecture Notes in Computer Science, vol. 7608, 2012, pp. 143-154.

[10] Lin and Chin-Yew, “ROUGE: A Package for Automatic Evaluation of Summaries,” Proceedings of the 19th International Conference on Computational Linguistics, Barcelona, Spain, 2004, pp. 74-81.

[11] H. Zha, “Generic Summarization and Keyphrase Extraction Using Mutual Reinforcement Principle and Sentence Clustering,” Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2002, pp. 113-120.

[12] T. Hoffman, “Probabilistic Latent Semantic Indexing,” Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 1999, pp. 50-57.

[13] D. Blei, A. Ng, and M. Jordan, “Latent Dirichlet Allocation,” Journal of Machine Learning Research, vol. 3, pp. 993–1022, 2003.

[14] C. Manning, and H. Schlutze, “Foundations of Statistical Natural Language Processing,” The MIT Press, 1999.

[15] J. Carbonell, and J. Goldstein, “The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries,” Proceedings of the 21st Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, 1998, pp. 335-336.

[16] J. Steinberger and K. Jezek, “Using Latent Semantic Analysis in Text Summarization and Summary Evaluation,” Proceedings of the International Conference on Information Systems Implementation and Modeling (ISIM) 2004, pp. 93-100.

[17] G. Murray, S. Renals and J. Carletta, “Extractive Summarization of Meeting Recordings,” Proceedings of Interspeech, Lisbon, Portugal, 2005, pp. 593-596.

[18] D. Mallett, J. Elding, and M. Nascimento, “Information-content Based Sentence Extraction for Text Summarization,” Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC), vol. 2, pp. 214, 2004.

[19] D. Radev, H. Jing, and M. Budzikowska, “Centroid-based Summarization of Multiple Documents: Sentence Extraction, Utility-based Evaluation, and User Studies,” Proceedings of the NAACL-ANLP Workshop on Automatic Summarization, vol. 4, pp. 21-30, 2000.

[20] T. Nomoto, and Y. Matsumoto, “A New Approach to Unsupervised Text Summarization,” Proceedings of the 24th International Conference on Research in Information Retrieval (SIGIR), pp. 26–34, 2001.

[21] D. Radev, E. Hovy, and K. McKeown, “Introduction to the Special Issue on Summarization,” Computational Linguistics, 28(4), pp. 399-408, 2002.

[22] J. Carletta, “Assessing Agreement on Classification Tasks: The Kappa Statistic,” Computational Linguistics, vol. 22, no. 2, pp. 249-254, 1996.

[23] D. Radev and D. Tam, “Summarization Evaluation Using Relative Utility,” Proceedings of the 12th International Conference on Information and Knowledge Management, pp. 508-511, 2003.

[24] K. Papineni, S. Roukos, T. Ward, and W. Zhu, “BLEU: a Method for Automatic Evaluation of Machine Translation,” Proceedings of the Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, pp. 311-318, 2002.